

УДК 004.42
ББК 32.973
А 45

Автор-составитель С. М. Мовшович, канд. техн. наук, доцент

Рецензенты: О. А. Кравченко, канд. физ.-мат. наук, доцент
Гомельского государственного технического
университета им. П. О. Сухого;
А. Н. Семенюта, д-р техн. наук, профессор,
зав. кафедрой информационно-вычислительных систем
Белорусского торгово-экономического университета
потребительской кооперации

Рекомендовано к изданию научно-методическим советом учреждения образования «Белорусский торгово-экономический университет потребительской кооперации». Протокол № 3 от 12 февраля 2013 г.

А 45 **Алгоритмизация** и программирование : пособие по выполнению курсовых работ для реализации содержания образовательных программ высшего образования I ступени / авт.-сост. С. М. Мовшович. – Гомель : учреждение образования «Белорусский торгово-экономический университет потребительской кооперации», 2014. – 56 с.
ISBN 978-985-540-173-6

В издании сформулированы цели и задачи курсовой работы, приведены варианты заданий, изложены требования к содержанию и оформлению курсовой работы, рассмотрен пример создания проекта.

Пособие предназначено для студентов специальности 1-26 03 01 «Управление информационными ресурсами».

УДК 004.42
ББК 32.973

ISBN 978-985-540-173-6

© Учреждение образования «Белорусский торгово-экономический университет потребительской кооперации», 2014

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Курсовая работа является одной из важных форм самостоятельной работы студентов специальности «Управление информационными ресурсами» при изучении учебной дисциплины «Алгоритмизация и программирование».

В процессе выполнения курсовой работы студент должен знать следующие теоретические аспекты использования интегрированной среды разработки Delphi:

- технологию работы с текстовыми файлами;
- методы создания многомодульных проектов и многостраничных форм;
- методы сортировки данных;
- приемы обработки данных различных типов;
- методику пошаговой отладки проекта;
- принципы формирования тестов;
- графические обозначения и стандарты представления схем алгоритмов;
- правила оформления программной документации.

При выполнении курсовой работы студент должен продемонстрировать умения и навыки:

- разработки функционально-модульной схемы проекта;
- проектирования интерфейса проекта;
- формирования теста в соответствии с условием решаемой задачи;
- выбора объектов формы и определения событий, для обработки которых требуется разработка процедур;
- разработки алгоритмов запланированных процедур;
- кодирования разработанных алгоритмов на языке среды Delphi;
- тестирования и отладки разработанного проекта;
- написание требуемой программной документации.

В пособии определены цели и содержание курсовой работы по дисциплине «Алгоритмизация и программирование», приведены варианты заданий, изложены требования к оформлению и описанию реализации проекта в среде Delphi 7.0. Подробно рассмотрен пример создания проекта.

1. ЦЕЛЬ, СОДЕРЖАНИЕ И ЗАЩИТА КУРСОВОЙ РАБОТЫ

Цель курсовой работы – создание программы (приложения) для обработки экономической информации на базе интегрированной среды разработки Delphi.

Содержание курсовой работы включает:

- создание текстового файла с исходными данными в соответствии с вариантом задания;
- подготовку теста с помощью табличного процессора Microsoft Excel;
- разработку функционально-модульной схемы проекта, состоящего из нескольких модулей с иерархическим меню или использующего многостраничную форму;
- разработку интерфейса каждого модуля или каждой страницы;
- определение набора процедур обработки событий;
- разработку алгоритмов и составление кода каждой процедуры;
- тестирование и отладку проекта.

Курсовая работа представляется на проверку вместе с электронной папкой, содержащей проект, исходный и выходной текстовые файлы и электронную книгу с тестом. Все материалы сдаются в методический кабинет кафедры информационно-вычислительных систем, где они регистрируются. Руководитель курсовой работы в срок до 7 дней проверяет представленные материалы и возвращает в методический кабинет курсовую работу с рецензией. В рецензии указываются замечания, отмечаются аспекты, требующие исправления, и делается вывод о допуске к защите.

На защиту студент представляет курсовую работу, дополненную листами с исправлениями, рецензию (все рецензии, если курсовая работа проверялась несколько раз) и проект с исправлениями.

2. СОДЕРЖАНИЕ И ОФОРМЛЕНИЕ КУРСОВОЙ РАБОТЫ

2.1. Структурные части курсовой работы

Курсовая работа должна содержать следующие структурные части:

- Титульный лист.
- Содержание.
- Постановка задачи. Тест.
- Функционально-модульная структура проекта.
- Интерфейс проекта.

- Описание алгоритмов и тексты процедур.
- Результат выполнения проекта.
- Инструкция пользователя.
- Направления развития проекта.
- Список использованных источников.
- Приложения.

2.2. Общие правила оформления

Набор текста курсовой работы осуществляется с использованием текстового редактора Microsoft Word и печатается на одной стороне листа белой бумаги формата А4 (210 × 297 мм).

При наборе необходимо использовать гарнитуру Times New Roman с размером шрифта 14 пунктов, установить межстрочный интервал 18 пунктов (при этом на странице будет 39–40 строк), отступ первой строки абзаца 12,5 мм и выравнивание по ширине. Расстановка переносов слов в тексте должна выполняться автоматически.

Устанавливаются следующие размеры полей: верхнего и нижнего – 20 мм, левого – 30, правого – 10 мм.

Разрешается использовать в тексте работы курсивное и полужирное начертание для акцентирования внимания на важных положениях текста. Опечатки разрешается исправлять подчисткой или корректором и нанесением на том же месте исправленного текста компьютерным или рукописным способом черной пастой.

Объем курсовой работы должен быть в пределах 20–40 страниц текста.

Заголовки структурных частей курсовой работы печатают прописными буквами по центру строк без абзацного отступа полужирным шрифтом.

Каждую структурную часть следует начинать с нового листа. Структурные части «Содержание», «Список использованных источников» и «Приложения» не нумеруются. Структурная часть «Приложения» может отсутствовать.

Нумерованные структурные части (разделы) при необходимости можно разбивать на подразделы (например, 1.1, 1.2 и т. д.) и пункты (например, 2.2.1, 2.2.2 и т. д.).

Заголовки подразделов и пунктов печатают полужирным шрифтом строчными буквами (первая прописная). Для заголовков устанавливается абзацный отступ 12,5 мм, выравнивание по левому краю. В конце заголовков точку не ставят. Если заголовок включает несколько

предложений, их разделяют точками. Перенос слов в заголовках, названиях таблиц и рисунков не допускается.

Нумерация структурных частей (разделов), подразделов, пунктов, рисунков, таблиц, формул, уравнений дается арабскими цифрами.

Подразделы нумеруют в пределах каждой главы. Номер подраздела состоит из номера раздела и порядкового номера подраздела, разделенных точкой (например, 3.2 – второй подраздел третьего раздела).

Пункты нумеруют в пределах каждого подраздела. Номер пункта состоит из порядковых номеров раздела, подраздела и пункта, разделенных точками (например, 2.4.1 – первый пункт четвертого подраздела второго раздела).

Заголовки разделов, подразделов и пунктов приводят после их номеров без точки через табуляцию, размер которой должен быть один и тот же для однотипных заголовков.

Интервал между заголовком и последующим текстом должен составлять 12 пунктов. Если между двумя заголовками текст отсутствует (например, между заголовками раздела и подраздела), то интервал между ними устанавливается в 12 пунктов. Интервал между заголовком и предшествующим текстом должен составлять 24 пункта.

Страницы работы следует нумеровать арабскими цифрами размером 12 пунктов с соблюдением сквозной нумерации по всему тексту курсовой работы. Номер страницы проставляется в центре нижнего колонтитула без точки. Титульный лист включают в общую нумерацию страниц, однако номер страницы на титульном листе не проставляют. Рисунки и таблицы, расположенные на отдельных листах, включают в общую нумерацию страниц курсовой работы. Рисунки и таблицы на листе формата А3 учитывают как одну страницу.

Иллюстрации (рисунки) и таблицы следует располагать в курсовой работе непосредственно после абзаца, в котором они упоминаются впервые. На все таблицы и иллюстрации должны быть ссылки в тексте курсовой работы.

Иллюстрации и таблицы обозначают словами «Рисунок» и «Таблица» соответственно и нумеруют последовательно в пределах каждого раздела.

Номер иллюстрации и таблицы должен состоять из номера раздела и порядкового номера иллюстрации (таблицы), разделенных точкой (например, «Рисунок 1.2» – второй рисунок первого раздела, «Таблица 2.5» – пятая таблица второго раздела). Если в каждом разделе курсовой работы приведено лишь по одной иллюстрации (таблице), то их нумеруют последовательно в пределах всей работы.

Иллюстрации, как правило, имеют наименование и пояснительные данные (подрисуночный текст), располагаемые по центру страницы. Пояснительные данные печатают размером шрифта 12 пунктов и помещают под иллюстрацией, а на следующей строке – слово «Рисунок», его номер и наименование иллюстрации, отделяя знаком тире номер от наименования. Точку после номера и наименования иллюстрации не ставят. Не допускается перенос слов в наименовании рисунка. Слово «Рисунок», его номер и наименование печатают по центру страницы полужирным шрифтом размером 12 пунктов.

Каждая таблица должна иметь краткий заголовок, который состоит из слова «Таблица», ее порядкового номера и названия, отделенного от номера знаком тире, и помещается над таблицей слева без абзацного отступа, печатается обычным шрифтом размером 12 пунктов.

Размер шрифта таблицы должен составлять 12 пунктов (головки таблицы – 10 пунктов), также должны соблюдаться одинарный межстрочный интервал и интервалы по 2 пункта перед текстом и после текста в ячейках таблицы.

Таблицу с большим количеством строк переносят на следующий лист таким образом, чтобы головка таблицы не была оторвана от строк. При переносе части таблицы на другой лист ее заголовок указывают один раз над первой частью, слева над другими (переносимыми) частями пишут слово «Продолжение», а затем «Окончание» и указывают номер таблицы (например, «Окончание таблицы 1.2»).

Заголовки граф и строк пишут с прописной буквы в единственном числе, а подзаголовки граф – со строчной, если они составляют одно предложение с заголовком, и с прописной, если они имеют самостоятельное значение.

Головка таблицы отделяется линией от остальной части таблицы. Слева, справа и снизу таблица также ограничивается линиями. Горизонтальные и вертикальные линии, разграничивающие строки и графы таблицы, могут не проводиться, если это не затрудняет чтение таблицы.

Приложения оформляют как продолжение курсовой работы, располагая их в порядке появления ссылок в тексте. Не допускается включение в приложения материалов, на которые отсутствуют ссылки в тексте работы. Перед приложениями на отдельном листе прописными буквами полужирным шрифтом печатают слово «Приложения». Каждое приложение следует начинать с нового листа, печатая в правом верхнем углу слово «Приложение» и его буквенное обозначение. Заголовок приложения размещается по центру новой строки без абзацного отступа с прописной буквы.

Приложения обозначают заглавными буквами русского алфавита, начиная с А (за исключением букв Ё, З, Й, О, Ч, Ъ, Ы, Ъ). После слова «Приложение» следует буква, обозначающая его последовательность (например, «Приложение А»).

2.3. Титульный лист

Пример оформления титульного листа приведен в приложении.

2.4. Составление оглавления курсовой работы

Содержание (оглавление) курсовой работы должно быть составлено с помощью функции Microsoft Word *Вставка/Оглавление и указатели*.

В содержание необходимо внести все заголовки разделов, подразделов и пунктов курсовой работы. Для этого заголовки должны быть отформатированы с использованием стилей.

2.5. Постановка задачи. Тест

В рассматриваемом подразделе курсовой работы необходимо привести формулировку полученного варианта задания и представить распечатки листов книги Excel (в виде рисунков) с примером заполнения таблицы и выполненными расчетами. Кроме того, надо привести в виде рисунка содержимое текстового файла, который будет использован для тестирования.

2.6. Функционально-модульная структура проекта

Рекомендуется для создаваемого проекта выбрать одну из двух структур:

- многомодульную (в которой каждому модулю соответствует одна форма);
- многостраничную (на основе единственной формы).

Вид структуры определяется студентом. Студент может предложить собственную структуру проекта, основанную на комбинации двух рекомендуемых.

При многомодульной структуре проекта на первую форму целесообразно поместить иерархическое меню, команды которого вызывают остальные формы. На одной из форм должно быть отражено содержимое исходного текстового файла, а на другой – результаты указанных в задании преобразований и вычислений. Обязательно должна присутствовать форма, отображающая сведения об авторе проекта. Назначение и содержимое остальных форм определяются студентом.

Если проект организован на основе многостраничной формы, то вызов страницы осуществляется с помощью вкладок. Содержимое страниц организуется по аналогии с указанным выше содержимым форм.

В данном подразделе следует привести функционально-модульную схему проекта, на которой отображаются связи между модулями (страницами) и выполняемые функции. Пример функционально-модульной схемы проекта будет рассмотрен в разделе 4.

2.7. Интерфейс проекта

Описание интерфейса проекта должно быть представлено рисунками, изображающими окна форм или страниц. Перед каждым рисунком надо указать назначение представленной формы (страницы) и условие ее отображения в процессе выполнения проекта.

Каждый рисунок должен представлять собой скриншот окна формы (страницы) с последующей обработкой в графическом редакторе для удаления лишних элементов. При необходимости можно добавить на скриншот поясняющие надписи.

В рассматриваемом подразделе курсовой работы для каждой формы необходимо перечислить используемые визуальные компоненты с указанием значений свойств, наиболее существенных для данной задачи.

2.8. Описание алгоритмов и тексты процедур

Состав процедур каждого модуля надо проиллюстрировать скриншотом интерфейсной части модуля. Для каждой процедуры необходимо указать ее назначение и событие, при котором она вызывается.

Алгоритм каждой процедуры должен быть описан словесно. Кроме того, для специфических процедур проекта (ввод данных из

текстового файла и заполнение рабочих массивов, обработка данных, запись результатов в новый файл) необходимо построить блок-схемы алгоритмов. Каждая блок-схема должна уместиться на одной странице. Если это условие не удастся выполнить, то блок-схему надо разделить на несколько по принципу детализации отдельных блоков.

Тексты процедур следует представлять в виде скриншотов исполняемой части модуля.

Текст каждой процедуры должен быть детально прокомментирован: в разделе описаний должны быть комментарии, поясняющие назначения всех используемых переменных, а в тексте процедуры комментарии должны сопровождать каждую логическую единицу алгоритма.

2.9. Результат выполнения проекта

Результат выполнения проекта для тестовых данных следует представить в виде скриншота окна формы (страницы) и распечатки содержимого сформированного текстового файла.

2.10. Инструкция пользователя

Инструкция пользователя должна включать следующие структурные элементы:

- Назначение программы.
- Условия выполнения программы.
- Выполнение программы.
- Сообщения пользователю.

Часть *Назначение программы* должна содержать сведения о назначении программы и информацию, достаточную для понимания функций программы и ее эксплуатации.

В части *Условия выполнения программы* должны быть указаны условия, необходимые для выполнения программы (минимальный и (или) максимальный состав аппаратурных и программных средств и т. п.).

В части *Выполнение программы* должна быть указана последовательность действий пользователя, обеспечивающих загрузку, запуск, выполнение и завершение программы. Кроме того, должно быть приведено описание доступных функций, формата исходных данных,

возможных вариантов команд, с помощью которых пользователь осуществляет загрузку и управляет выполнением программы.

Часть *Сообщения пользователю* должна содержать тексты сообщений, выдаваемых в ходе выполнения программы, описание их содержания и соответствующие действия пользователя (действия пользователя в случае сбоя, возможности повторного запуска программы, возможность исправления последовательности действий и т. п.).

2.11. Направления развития проекта

В этом подразделе автор может описать функции, по какой-то причине не включенные в проект, но которые могли бы повысить эффективность его использования.

2.12. Список использованных источников

В работе приводится список литературных источников, использованных при ее выполнении и написании, а также источников, на которые есть ссылки. В данный список включается программно-техническая документация, а также учебно-методическая литература. Список может включать ссылки на интернет-источники.

При оформлении списка использованных источников необходимо пользоваться образцами, приведенными в списке рекомендуемой литературы данного пособия.

2.13. Приложения

Приложения – необязательная структурная часть курсовой работы. В приложения к курсовой работе можно поместить дополнительные тесты и результаты выполнения программы для этих данных, различные версии интерфейса, варианты схем алгоритмов и текста процедур и т. д.

3. ВАРИАНТЫ ЗАДАНИЙ

Вариант 1

Автоматизация учета комплектации выпускаемых цехами изделий на основе текстового файла

В текстовом файле находится таблица, содержащая сведения о комплектации выпускаемых цехами изделий. Таблица состоит из четырех граф: *Цех; Наименование изделия; Наименование детали; Количество деталей, шт.* Известно, что одна и та же деталь может входить в разные изделия и изделие может состоять из разных деталей. Разные цеха могут изготавливать одни и те же детали.

Разработайте алгоритм и программу подсчета количества одноименных деталей, изготовленных всеми цехами для всех изделий. Результаты расчета запишите в новый текстовый файл, содержащий таблицу из двух граф: *Наименование детали; Общее количество деталей, шт.*

Вариант 2

Автоматизация учета поставок товаров на основе текстового файла

В текстовом файле находится ведомость, содержащая данные о поставках товаров и состоящая из пяти граф: *Наименование товара; Дата поставки; Единица измерения; Цена, р.; Количество.* Известно, что в один день может быть выполнено несколько поставок, в том числе одного и того же товара.

Разработайте алгоритм и программу вычисления общей стоимости поставок каждого товара. Результаты расчета запишите в новый текстовый файл, содержащий таблицу из двух граф: *Наименование товара; Стоимость, р.*

Вариант 3

Автоматизация учета простоев конвейера на основе текстового файла

В текстовом файле находится сводная ведомость о простоях конвейера с начала месяца по некоторому цеху, содержащая пять граф: *Дата, Причина простоя, 1-я смена, 2-я смена, 3-я смена.* В последних трех графах указано время простоя в часах.

Разработайте алгоритм и программу определения времени простоя в каждый день месяца по всем причинам за все смены. Результаты расчета запишите в новый текстовый файл, содержащий таблицу из двух граф: *Дата; Общее время простоя, ч.*

Вариант 4

Автоматизация учета стоимости выпущенной продукции по кварталам на основе текстового файла

В текстовом файле находится таблица, содержащая сведения о стоимости выпущенной продукции по кварталам и состоящая из шести граф: *Наименование предприятия, Наименование изделия, Квартал I, Квартал II, Квартал III, Квартал IV*. В последних четырех графах указана стоимость выпущенной продукции (тыс. р.) в соответствующем квартале. Известно, что одно и то же изделие может выпускаться разными предприятиями и каждое предприятие может выпускать несколько видов изделий.

Разработайте алгоритм и программу вычисления выпуска продукции в денежном выражении каждым предприятием за год по всем изделиям. Результаты расчета запишите в новый текстовый файл, содержащий таблицу из двух граф: *Наименование предприятия; Годовой объем выпуска, тыс. р.*

Вариант 5

Автоматизация учета продажи товаров на основе текстового файла

В текстовом файле находится ведомость, содержащая данные о продаже товаров и состоящая из пяти граф: *Наименование товара; Дата продажи; Отпускная цена производителя, р.; Торговая надбавка, %; Количество*. Известно, что в один день может быть продано несколько видов товара.

Разработайте алгоритм и программу вычисления общей стоимости дохода от продажи каждого товара за весь период учета. Результаты расчета запишите в новый текстовый файл, содержащий таблицу из двух граф: *Наименование товара; Стоимость, р.*

Вариант 6

Автоматизация обработки технологических карт выпускаемых изделий на основе текстового файла

В текстовом файле находится технологическая карта, содержащая данные на каждое из выпускаемых изделий и состоящая из пяти граф: *Наименование изделия; Наименование детали; Количество деталей в изделии; Наименование материала; Норма расхода материала на одну деталь, кг*. Известно, что в одно изделие могут входить несколько деталей и на изготовление одной детали могут потребоваться различные материалы.

Разработайте алгоритм и программу определения потребности каждого материала на все изделия. Результаты расчета запишите в новый текстовый файл, содержащий таблицу из двух граф: *Наименование материала; Потребность, кг*.

Вариант 7

Автоматизация определения динамики изменения стоимости поставок товаров на основе текстового файла

В текстовом файле находится ведомость, содержащая данные о поставках товаров и состоящая из пяти граф: *Наименование товара; Дата поставки; Единица измерения; Цена, р.; Количество*. Известно, что в один день может быть выполнено несколько поставок, в том числе одного и того же товара.

Разработайте алгоритм и программу вычисления общей стоимости поставок каждого товара. Результаты расчета запишите в новый текстовый файл, содержащий таблицу из двух граф: *Дата поставки; Стоимость, р*.

Вариант 8

Автоматизация учета ежемесячных доходов и затрат предприятий на основе текстового файла

В текстовом файле находится таблица, содержащая сведения о ежемесячных доходах и затратах предприятий и состоящая из пяти граф: *Месяц; Год; Наименование предприятия; Затраты, млн р.; Доход, млн р*.

Разработайте алгоритм и программу вычисления прибыли каждого предприятия за весь учетный период. Результаты расчета запишите в новый текстовый файл, содержащий таблицу из двух граф: *Наименование предприятия; Прибыль, млн р*.

Вариант 9

Автоматизация учета ежемесячных сведений о количестве уволившихся работников на основе текстового файла

В текстовом файле находится таблица, содержащая ежемесячные сведения о количестве уволившихся работников и состоящая из пяти граф: *Подразделение предприятия, Месяц, Год, Категория работников, Число уволившихся*. Рассматриваются четыре категории работников: рабочие, руководители, специалисты и технические исполнители (служащие).

Разработайте алгоритм и программу вычисления общего числа уволившихся работников каждой категории за весь период учета. Результаты расчета запишите в новый текстовый файл, содержащий таблицу из двух граф: *Категория работников, Общее число уволившихся*.

Вариант 10

Автоматизация учета поставок товаров от производителей на основе текстового файла

В текстовом файле находится таблица, содержащая сведения о поставках товаров от производителей и состоящая из пяти граф: *Производитель; Наименование товара; Дата поставки; Объем поставки, шт.; Отпускная цена производителя, р.* Известно, что один и тот же товар поставлялся разными производителями и один производитель поставлял разные товары.

Разработайте алгоритм и программу вычисления средневзвешенной цены каждого товара за весь период учета. Результаты расчета запишите в новый текстовый файл, содержащий таблицу из двух граф: *Наименование товара; Средневзвешенная цена, р.*

Вариант 11

Автоматизация вычисления баланса по каждой валюте для контрагентов на основе текстового файла

В текстовом файле находится таблица, содержащая сведения о валютных операциях организации и состоящая из пяти граф: *Дата, Контрагент, Валюта, Сумма выплаты, Сумма поступления*.

Разработайте алгоритм и программу вычисления баланса по каждой валюте за весь период учета для всех контрагентов. Результаты

расчета запишите в новый текстовый файл, содержащий таблицу из двух граф: *Валюта, Баланс*.

Вариант 12

Автоматизация вычисления общей суммы расходов по каждому подразделению на основе текстового файла

В текстовом файле находится таблица, содержащая сведения о ежедневных расходах организации по подразделениям. Таблица состоит из шести граф: *Дата; Подразделение; Статья расходов; Единица измерения; Тариф, р.; Объем*. Примеры статей расходов: электроэнергия, тепловая энергия, заработная плата и т. д.

Разработайте алгоритм и программу вычисления общей суммы расходов по каждому подразделению за весь период учета. Результаты расчета запишите в новый текстовый файл, содержащий таблицу из двух граф: *Подразделение; Сумма расходов, р.*

Вариант 13

Автоматизация определения динамики стоимости доставок товаров в магазины на основе текстового файла

В текстовом файле находится таблица, содержащая сведения о доставках товара в магазины и состоящая из пяти граф: *Дата; Магазин; Расстояние до магазина, км; Тариф за 1 т·км, р.; Объем, т.*

Разработайте алгоритм и программу вычисления общей стоимости расходов на доставку товаров по всем магазинам за каждый день. Результаты расчета запишите в новый текстовый файл, содержащий таблицу из двух граф: *Дата; Стоимость доставки, р.*

Вариант 14

Автоматизация ведения операций на расчетном счете организации на основе текстового файла

В текстовом файле находится таблица, содержащая сведения об операциях на расчетном счете организации и состоящая из четырех граф: *Дата; Контрагент; Операция; Сумма, р.* Рассматриваются два вида операций: поступление и выплата.

Разработайте алгоритм и программу вычисления общей суммы поступлений, общей суммы выплат и баланса для каждого контрагента за весь период учета. Результаты расчета запишите в новый тексто-

вый файл, содержащий таблицу из четырех граф: *Контрагент*; *Сумма поступлений, р.*; *Сумма выплат, р.*; *Баланс, р.*

Вариант 15

Автоматизация учета годового выпуска деталей на основе текстового файла

В текстовом файле находится таблица, содержащая сведения об объеме выпуска деталей по кварталам и состоящая из шести граф: *Наименование изделия*; *Наименование детали*; *Квартал I*; *Квартал II*; *Квартал III*; *Квартал IV*. В последних четырех графах указан объем выпущенной продукции (шт.) в соответствующем квартале. Известно, что одна и та же деталь может входить в разные изделия и каждое изделие может содержать несколько видов деталей.

Разработайте алгоритм и программу вычисления выпуска каждой детали за год. Результаты расчета запишите в новый текстовый файл, содержащий таблицу из двух граф: *Наименование детали*; *Годовой объем выпуска, шт.*

Вариант 16

Автоматизация учета покупки валюты организациями на основе текстового файла

В текстовом файле находится таблица, содержащая сведения о покупке валюты организациями и состоящая из пяти граф: *Дата*; *Наименование организации*; *Валюта*; *Сумма покупки, бел. р.*; *Курс, бел. р.*

Разработайте алгоритм и программу вычисления общей суммы покупки в рублях для каждой организации за весь период учета. Результаты расчета запишите в новый текстовый файл, содержащий таблицу из двух граф: *Наименование организации*; *Сумма покупки, бел. р.*

Вариант 17

Автоматизация определения процента выполнения плана выпуска деталей на основе текстового файла

В текстовом файле находится таблица, содержащая сведения о выпуске деталей разными цехами предприятия и состоящая из пяти граф: *Дата*; *Цех*; *Наименование детали*; *План выпуска, шт.*; *Факти-*

чески выпущено, шт. Известно, что разные цеха могут изготавливать одни и те же детали.

Разработайте алгоритм и программу вычисления процента выполнения плана по каждой детали за весь период всеми цехами. Результаты расчета запишите в новый текстовый файл, содержащий таблицу из двух граф: *Наименование детали; Выполнение плана, %*.

Вариант 18

Автоматизация учета стоимости ремонта бытовой техники на основе текстового файла

В текстовом файле находится таблица, содержащая сведения о ремонтах бытовой техники и состоящая из пяти граф: *Дата; Наименование бытовой техники; Вид ремонта; Тариф, р.; Срочность, %*. Значениями последней графы могут быть 10% или 0.

Разработайте алгоритм и программу вычисления дохода от ремонта каждого типа бытовой техники за весь период учета для всех видов ремонта. Результаты расчета запишите в новый текстовый файл, содержащий таблицу из двух граф: *Наименование бытовой техники; Доход, р.*

Вариант 19

Автоматизация вычисления постатейной суммы расходов на основе текстового файла

В текстовом файле находится таблица, содержащая сведения о ежедневных расходах организации по подразделениям. Таблица состоит из шести граф: *Дата; Подразделение; Статья расходов; Единица измерения; Тариф, р.; Объем*. Примеры статей расходов: электроэнергия, тепловая энергия, заработная плата и т. д.

Разработайте алгоритм и программу вычисления общей суммы расходов по каждой статье за весь период учета. Результаты расчета запишите в новый текстовый файл, содержащий таблицу из двух граф: *Статья расходов; Сумма расходов, р.*

Вариант 20

Автоматизация определения стоимости поставок товарных групп на основе текстового файла

В текстовом файле находится ведомость, содержащая данные о поставках товаров и состоящая из пяти граф: *Наименование товарной группы; Наименование товара; Дата поставки; Цена, р.; Количество, кг.* Известно, что в один день может быть выполнено несколько поставок, в том числе одного и того же товара и товаров из одной группы.

Разработайте алгоритм и программу вычисления общей стоимости поставок товаров каждой товарной группы. Результаты расчета запишите в новый текстовый файл, содержащий таблицу из двух граф: *Наименование товарной группы; Стоимость, р.*

Вариант 21

Автоматизация учета задолженности по договорам поставок на основе текстового файла

В текстовом файле находится таблица, содержащая сведения о выплатах фирмы по договорам поставок и состоящая из пяти граф: *Номер договора; Контрагент; Сумма по договору, р.; Задолженность, дней; Пеня за 1 день, %.* С одним контрагентом может быть заключено несколько договоров.

Разработайте алгоритм и программу вычисления необходимой суммы выплат по каждому контрагенту по всем договорам. Результаты расчета запишите в новый текстовый файл, содержащий таблицу из двух граф: *Контрагент; Общая сумма выплат, р.*

Вариант 22

Автоматизация учета выплат сотрудникам организации на основе текстового файла

В текстовом файле находится таблица, содержащая сведения о выплатах сотрудникам организации и состоящая из пяти граф: *Подразделение; ФИО; Оклад, р.; Доплата, %; Отчисления, %.* Значения последних двух граф указаны в процентах к окладу.

Разработайте алгоритм и программу вычисления фонда заработной платы по каждому подразделению. Результаты расчета запишите в новый текстовый файл, содержащий таблицу из двух граф: *Подразделение; Фонд заработной платы, р.*

Вариант 23

Автоматизация учета движения материалов на складах фирмы на основе текстового файла

В текстовом файле находится таблица, содержащая сведения о движении материалов на складах фирмы. Таблица состоит из пяти граф: *Дата*; *Номер склада*; *Наименование материала*; *Приход, т*; *Расход, т*. Один и тот же материал может поступать на разные склады.

Разработайте алгоритм и программу вычисления общего объема каждого материала, находящегося на всех складах за весь период учета, при условии, что первоначально склады были пустые. Результаты расчета запишите в новый текстовый файл, содержащий таблицу из двух граф: *Наименование материала*; *Остаток, т*.

Вариант 24

Автоматизация учета выпуска экспортных изделий на основе текстового файла

В текстовом файле находится таблица, содержащая сведения о выпуске изделий на экспорт и состоящая из пяти граф: *Дата*; *Наименование предприятия*; *Наименование изделия*; *Выпущено, млн р.*; *В том числе на экспорт, млн р.* Известно, что разные предприятия могут выпускать одни и те же изделия.

Разработайте алгоритм и программу вычисления доли экспорта в общем доходе предприятия за весь период учета. Результаты расчета запишите в новый текстовый файл, содержащий таблицу из двух граф: *Наименование предприятия*; *Доля экспорта в доходе, %*.

Вариант 25

Автоматизация учета поступления на склады годных товаров на основе текстового файла

В текстовом файле находится таблица, содержащая сведения о поступлениях товара на склады фирмы и состоящая из пяти граф: *Дата*; *Номер склада*; *Наименование товара*; *Поступило по накладной, т*; *Брак, %*. Приходные документы оформляются только на годный товар.

Разработайте алгоритм и программу вычисления общего объема каждого товара, оприходованного на складах за весь период учета. Результаты расчета запишите в новый текстовый файл, содержащий таблицу из двух граф: *Наименование товара*; *Оприходовано, т*.

Вариант 26

Автоматизация учета месячного выпуска деталей на основе текстового файла

В текстовом файле находится таблица, содержащая сведения о выпуске деталей разными цехами предприятия. Таблица состоит из пяти граф: *Месяц; Цех; Наименование детали; План выпуска, шт.; Выполнение плана, %*. Известно, что разные цеха могут изготавливать одни и те же детали.

Разработайте алгоритм и программу вычисления фактического выпуска всех деталей в каждый месяц всеми цехами. Результаты расчета запишите в новый текстовый файл, содержащий таблицу из двух граф: *Месяц; Фактически выпущено, шт.*

Вариант 27

Автоматизация определения производительности труда рабочих на основе текстового файла

В текстовом файле находится таблица, содержащая сведения о производительности труда рабочих предприятий и состоящая из пяти граф: *Дата; Наименование предприятия; Наименование изделия; Число рабочих; Фактически выпущено, шт.* Известно, что разные предприятия могут выпускать одни и те же изделия.

Разработайте алгоритм и программу вычисления средней производительности труда (количество изделий на одного рабочего) для каждого изделия всеми предприятиями за весь период. Результаты расчета запишите в новый текстовый файл, содержащий таблицу из двух граф: *Наименование изделия; Производительность труда, шт.*

Вариант 28

Автоматизация определения стоимости доставок товаров по магазинам на основе текстового файла

В текстовом файле находится таблица, содержащая сведения о доставках товара в магазины и состоящая из пяти граф: *Дата; Магазин; Расстояние до магазина, км; Тариф за 1 т·км, р.; Объем, т.*

Разработайте алгоритм и программу вычисления общей стоимости расходов на доставку товаров по каждому магазину за весь период учета. Результаты расчета запишите в новый текстовый файл, содержащий таблицу из двух граф: *Магазин; Стоимость доставки, р.*

Вариант 29

Автоматизация определения доли товарооборота импортных товаров по магазинам на основе текстового файла

В текстовом файле находится таблица, содержащая сведения о продажах товаров и состоящая из четырех граф: *Дата*; *Магазин*; *Товарооборот отечественных товаров, млн р.*; *Товарооборот импортных товаров, млн р.*

Разработайте алгоритм и программу вычисления доли импортных товаров в общем товарообороте каждого магазина за весь период учета. Результаты расчета запишите в новый текстовый файл, содержащий таблицу из трех граф: *Магазин*; *Общий товарооборот, млн р.*; *Доля импортных товаров, %*.

Вариант 30

Автоматизация определения средней заработной платы по организациям на основе текстового файла

В текстовом файле находится таблица, содержащая данные о фонде заработной платы организаций. Таблица состоит из пяти граф: *Месяц*; *Год*; *Наименование организации*; *Число работников*; *Фонд заработной платы, р.*

Разработайте алгоритм и программу вычисления средней заработной платы организации за весь период учета. Результаты расчета запишите в новый текстовый файл, содержащий таблицу из двух граф: *Наименование организации*; *Средняя заработная плата, р.*

4. ПРИМЕР ВЫПОЛНЕНИЯ ПРОЕКТА

4.1. Постановка задачи. Тест

В текстовом файле находится ведомость, содержащая данные о поставках товаров: *Наименование товара*; *Наименование поставщика*; *Единица измерения*; *Цена, р.*; *Количество*. Известно, что один поставщик может поставлять несколько разных товаров, и каждый товар может быть неоднократно поставлен разными фирмами.

Необходимо разработать алгоритм и программу вычисления общей стоимости поставок каждого поставщика, результаты расчета за-

писать в новый текстовый файл, содержащий таблицу из двух граф: *Наименование поставщика; Стоимость, р.*

На рисунке 1 приведена электронная таблица с подготовленными исходными данными.

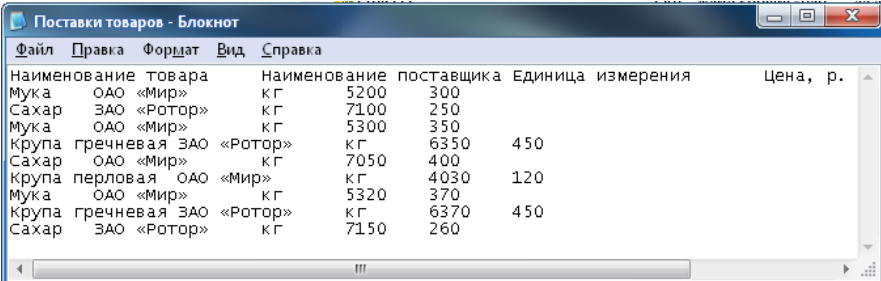
	А	В	С	Д	Е
1	Наименование товара	Наименование поставщика	Единица измерения	Цена, р.	Количество
2	Мука	ОАО «Мир»	кг	5200	300
3	Сахар	ЗАО «Ротор»	кг	7100	250
4	Мука	ОАО «Мир»	кг	5300	350
5	Крупа гречневая	ЗАО «Ротор»	кг	6350	450
6	Сахар	ОАО «Мир»	кг	7050	400
7	Крупа перловая	ОАО «Мир»	кг	4030	120
8	Мука	ОАО «Мир»	кг	5320	370
9	Крупа гречневая	ЗАО «Ротор»	кг	6370	450
10	Сахар	ЗАО «Ротор»	кг	7150	260

Рисунок 1 – Электронная таблица с исходными данными

С помощью команды *Сохранить как* сохраним исходные данные в новый файл *Поставки товаров*, в качестве типа которого укажем *Текстовые файлы (с разделителями табуляции)*. Текстовый файл сохраним в той папке, в которой будет находиться разрабатываемый проект.

Обратим внимание, что в исходных данных есть товар, поставляемый разными организациями, и есть поставщик, работающий с несколькими товарами.

Текстовый файл с исходными данными проекта приведен на рисунке 2. Отметим, что элементы данных в одной строке разделяются табуляцией.



Наименование товара	наименование поставщика	Единица измерения	Цена, р.
Мука	ОАО «Мир»	кг	5200
Сахар	ЗАО «Ротор»	кг	7100
Мука	ОАО «Мир»	кг	5300
Крупа гречневая	ЗАО «Ротор»	кг	6350
Сахар	ОАО «Мир»	кг	7050
Крупа перловая	ОАО «Мир»	кг	4030
Мука	ОАО «Мир»	кг	5320
Крупа гречневая	ЗАО «Ротор»	кг	6370
Сахар	ЗАО «Ротор»	кг	7150

Рисунок 2 – Текстовый файл *Поставки товаров.txt*

В электронной книге Excel создадим копию листа с исходными данными. Добавим столбец *Стоимость*, который заполним формулами для вычисления стоимости поставленных товаров в каждой строке. Выполним сортировку по полю *Наименование поставщика* и вставим строки с промежуточными итогами для вычисления общей стоимости товаров каждого поставщика. Таким образом, подготовлен тест для нашего проекта (рисунок 3), который сохраним под именем *Тест.xls* в той папке, в которой будет находиться разрабатываемый проект.

	А	В	С	Д	Е	Ф
1	Наименование товара	Наименование поставщика	Единица измерения	Цена, р.	Количество	Стоимость, р.
2	Сахар	ЗАО «Ротор»	кг	7100	250	1775000
3	Крупа гречневая	ЗАО «Ротор»	кг	6350	450	2857500
4	Крупа гречневая	ЗАО «Ротор»	кг	6370	450	2866500
5	Сахар	ЗАО «Ротор»	кг	7150	260	1859000
6		ЗАО «Ротор» Итог				9358000
7	Мука	ОАО «Мир»	кг	5200	300	1560000
8	Мука	ОАО «Мир»	кг	5300	350	1855000
9	Сахар	ОАО «Мир»	кг	7050	400	2820000
10	Крупа перловая	ОАО «Мир»	кг	4030	120	483600
11	Мука	ОАО «Мир»	кг	5320	370	1968400
12		ОАО «Мир» Итог				8687000
13		Общий итог				18045000

Рисунок 3 – Тест, подготовленный в табличном процессоре Microsoft Excel

4.2. Функционально-модульная структура проекта

Разработаем многомодульный проект с главным меню в окне первой формы *Form1*. Меню будет включать следующие команды: *Ввод исходных данных*, *Сортировка*, *Обработка*, *Об авторе*, *Выход*.

Модуль *Unit1*, соответствующий первой форме *Form1*, должен обеспечивать вызов остальных модулей и отображение соответствующих форм. Кроме того, перед началом работы (до выбора пользователем одной из команд меню) в модуле *Unit1* надо уничтожить старую версию выходного файла и установить признаки того, что ни одна из команд меню еще не выполнялась.

Реакция на выбор пользователем команды *Выход*, реализуемая в этом же модуле *Unit1*, должна включать контроль записи результатов в новый файл. Если запись не была выполнена, то выход из программы не должен осуществляться.

По команде *Ввод исходных данных* должно открываться окно второй формы *Form2*, при активации которой будут вводиться данные из текстового файла. Эти данные будут записываться в массивы и отображаться в таблице строк. Здесь же надо установить признак успешной обработки исходного файла. На форме должна находиться кнопка, обеспечивающая возврат к первой форме. Все перечисленные действия будут реализовываться в модуле *Unit2*.

Команда *Сортировка* должна вызывать третью форму *Form3*, при активации которой будет выполняться сортировка исходных массивов по полю *Наименование поставщика*, и результат сортировки отображаться в таблице строк. Сортировка должна выполняться только при наличии признака успешного ввода исходных данных. После выполнения сортировки программа должна установить соответствующий признак. На форме должна находиться кнопка, обеспечивающая возврат к первой форме. Все перечисленные действия будут реализовываться в модуле *Unit3*.

По команде *Обработка* должна вызываться четвертая форма *Form4*, при активации которой для каждого поставщика будет вычисляться общая стоимость всех поставленных товаров, результаты записываться в новые массивы и отображаться в таблице строк. Обработка массивов должна производиться только при наличии признаков успешного ввода исходных данных и выполнения сортировки. На форме должны находиться две командные кнопки. Одна из них будет вызывать процедуру записи результата вычислений в новый текстовый файл, а другая – обеспечивать возврат к первой форме. После успешного создания выходного файла программа должна установить соответствующий признак. Все перечисленные действия будут реализовываться в модуле *Unit4*.

Команда *Об авторе* должна вызывать окно пятой формы *Form5*, в которой будут отображены сведения об авторе. На форме должна находиться кнопка, обеспечивающая возврат к первой форме. Форме *Form5* будет соответствовать модуль *Unit5*.

Таким образом, проект будет содержать пять форм и, соответственно, пять модулей.

Описанная функциональная структура проекта приведена на рисунке 4.



Рисунок 4 – Функционально-модульная схема проекта

4.3. Интерфейс проекта

Вид формы *Form1* приведен на рисунке 5.

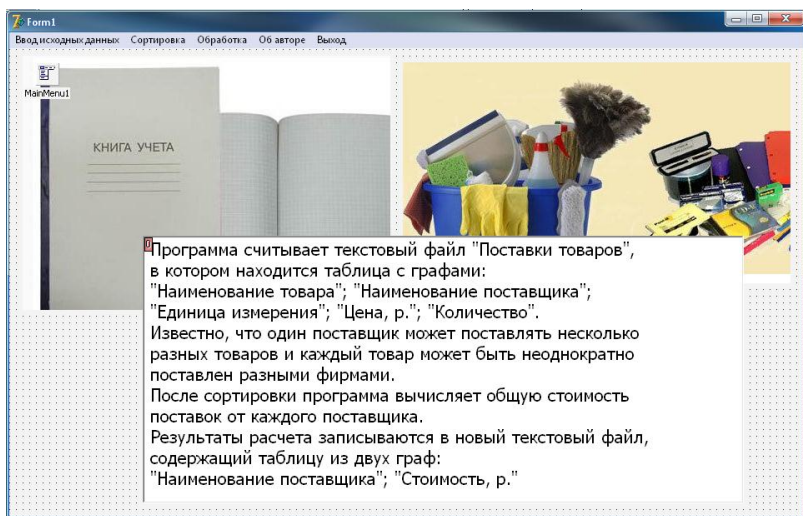


Рисунок 5 – Окно *Form1*

На форму помещено четыре визуальных компонента:

1. Компонент *MainMenu1* с пятью командами:

- *N1* – Ввод исходных данных;
- *N2* – Сортировка;
- *N3* – Обработка;
- *N4* – Об авторе;
- *N5* – Выход.

2. Компонент *Memo1*, содержащий строки с условием задачи.

3. Два компонента *Image1* и *Image2*, содержащие фоновые рисунки.

Значения свойств всех компонентов (кроме перечисленных выше) оставлены по умолчанию.

На форме *Form2* помещено четыре визуальных компонента:

1. Компонент *Label1*, в свойстве *Caption* которого установлено значение *Поставки товаров*.

2. Таблица строк *StringGrid1* со следующими значениями свойств:

- *ColCount* – 5;
- *RowCount* – 30;
- *FixedCols* – 0;

- *FixedRows* – 1;
 - *goEditing* – *False*.
3. Командная кнопка *Button1*, свойство *Caption* которой имеет значение *В главное меню*.
4. Компонент *Image1*, содержащий фоновый рисунок.
- Вид формы *Form2* приведен на рисунке 6.

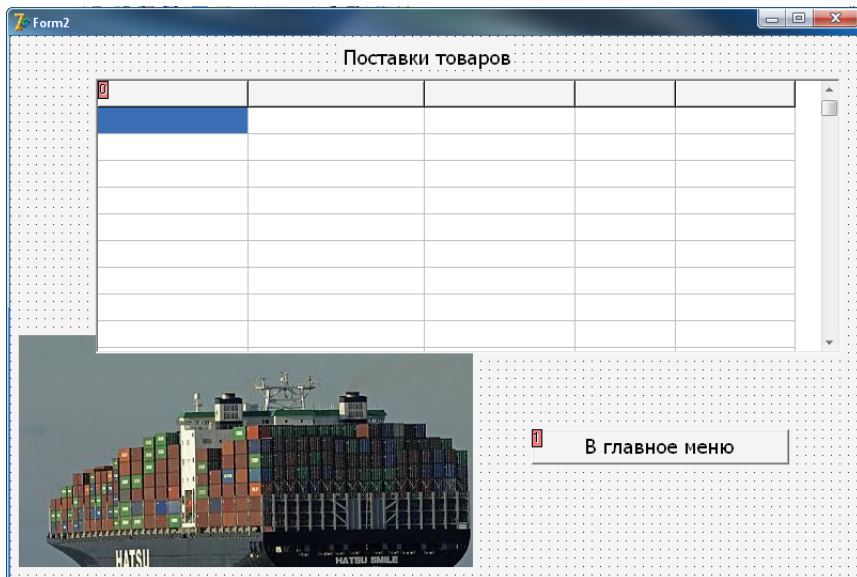


Рисунок 6 – Окно *Form2*

На форме *Form3* помещено четыре визуальных компонента:

1. Компонент *Label1*, в свойстве *Caption* которого установлено значение *Сортировка исходного файла по алфавиту поставщиков*.
 2. Таблица строк *StringGrid1* со следующими значениями свойств:
 - *ColCount* – 5;
 - *RowCount* – 30;
 - *FixedCols* – 0;
 - *FixedRows* – 1;
 - *goEditing* – *False*.
 3. Командная кнопка *Button1*, свойство *Caption* которой имеет значение *В главное меню*.
 4. Компонент *Image1*, содержащий фоновый рисунок.
- Вид формы *Form3* приведен на рисунке 7.

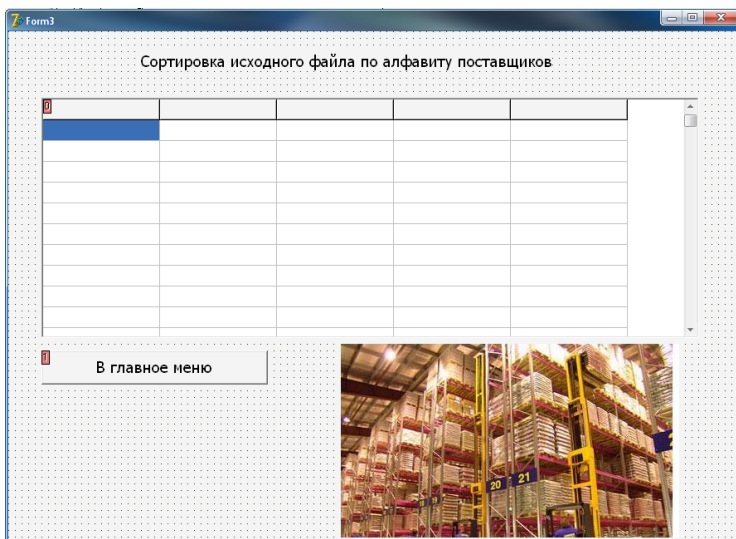


Рисунок 7 – Окно *Form3*

Вид формы *Form4* приведен на рисунке 8.

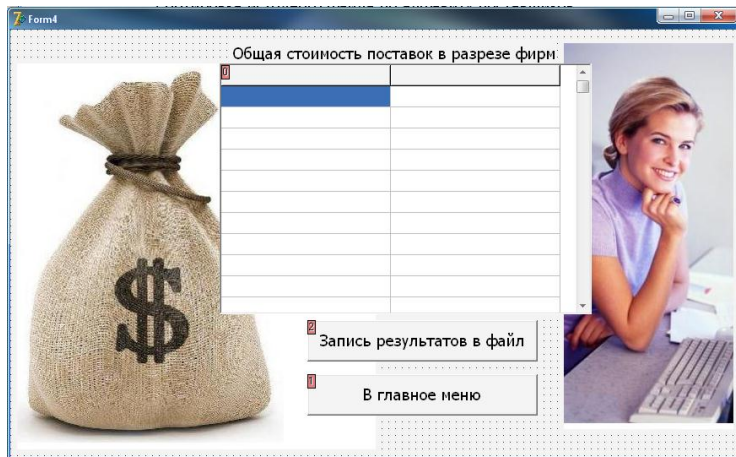


Рисунок 8 – Окно *Form4*

На форме *Form4* помещено шесть визуальных компонентов:

1. Компонент *Label1*, в свойстве *Caption* которого установлено значение *Общая стоимость поставок в разрезе фирм*.

2. Таблица строк *StringGrid1* со следующими значениями свойств:
 - *ColCount* – 2;
 - *RowCount* – 30;
 - *FixedCols* – 0;
 - *FixedRows* – 1;
 - *goEditing* – *False*.
 3. Командная кнопка *Button1*, свойство *Caption* которой имеет значение *Запись результатов в файл*.
 4. Командная кнопка *Button2*, свойство *Caption* которой имеет значение *В главное меню*.
 5. Два компонента *Image1* и *Image2*, содержащие фоновые рисунки.
- Структура окна формы *Form5* определяется автором самостоятельно.

4.4. Описания алгоритмов и тексты процедур

4.4.1. Модуль *Unit1*

Фрагмент интерфейсной части модуля *Unit1* и описание связи с другими модулями приведены на рисунке 9.

```
var
  Form1: TForm1;
  tov:array[1..30] of string[255]; //Массив наименований товаров
  post:array[1..30] of string[255]; //Массив наименований поставщиков
  ediz:array[1..30] of string[255]; //Массив единиц измерения
  cena:array[1..30] of Real; //Массив цен
  kol:array[1..30] of Real; //Массив объемов поставок
  n:Integer; //Число строк в исходном файле
  W1:Boolean; //Признак выполнения команды "Ввод исходных данных"
  W2:Boolean; //Признак выполнения команды "Сортировка"
  W3:Boolean; //Признак выполнения команды "Запись результатов в файл"
  fv2: TextFile; //Файловая переменная для создаваемого файла
implementation
  uses Unit2, Unit3, Unit4, Unit5;
```

Рисунок 9 – Фрагмент интерфейсной части модуля *Unit1*

Как видно из рисунка 9, в качестве глобальных используются пять массивов. Три из них (*tov*, *post*, *ediz*) имеют строковый тип и предназначены для хранения наименований товаров, наименований поставщиков и единиц измерения, вводимых из заданного текстового файла. Цены и количество поставленных товаров будут храниться в двух вещественных массивах (*cena*, *kol*). Элементы всех массивов получают значения в модуле *Unit2* и используются в модулях *Unit3* и *Unit4*.

Кроме указанных массивов, глобальными объявлены следующие переменные:

- целая переменная n (число строк в исходном файле без учета первой строки с заголовками граф) – получает значение в модуле *Unit2* и используется в модулях *Unit3* и *Unit4*;
- логическая переменная $W1$ – в модуле *Unit1* получает начальное значение *False*, в модуле *Unit2* получает значение *True* после успешного выполнения команды *Ввод исходных данных*, используется в модулях *Unit3* и *Unit4*;
- логическая переменная $W2$ – в модуле *Unit1* получает начальное значение *False*, в модуле *Unit3* получает значение *True* после успешного выполнения команды *Сортировка*, используется в модуле *Unit4*;
- логическая переменная $W3$ – в модуле *Unit1* получает начальное значение *False*, в модуле *Unit4* получает значение *True* после успешного выполнения команды *Запись результатов в файл*, используется в модуле *Unit1*;
- файловая переменная $fv2$ – получает значение в модуле *Unit1* и используется в модулях *Unit1* и *Unit4*.

Модуль *Unit1* содержит инициализирующую часть, которая приведена на рисунке 10.

```
initialization
W1:=False; //Команда "Ввод исходных данных" еще не выполнялась
W2:=False; //Команда "Сортировка" еще не выполнялась
W3:=False; //Команда "Запись результатов в файл" еще не выполнялась
{В случае существования файла "Общая стоимость поставок.txt"
 он уничтожается}
AssignFile(fv2,'Общая стоимость поставок.txt');
if FileExists('Общая стоимость поставок.txt')
then
    Erase(fv2);
end.
```

Рисунок 10 – Инициализирующая часть модуля *Unit1*

В инициализирующей части логические переменные $W1$, $W2$ и $W3$ получают начальные значения *False* как признаки того, что еще не выполнялись соответствующие команды меню. Кроме того, в рассматриваемой части модуля уничтожается старая версия файла *Общая стоимость поставок.txt*, если она существовала.

Исполняемая часть модуля *Unit1* содержит описания пяти процедур, которые обрабатывают соответствующие команды меню. Их тексты приведены на рисунке 11.

```

procedure TForm1.N1Click(Sender: TObject);
begin
    Form2.Show //Ввод исходных данных
end;

procedure TForm1.N2Click(Sender: TObject);
begin
    Form3.Show //Сортировка
end;

procedure TForm1.N3Click(Sender: TObject);
begin
    Form4.Show //Обработка
end;

procedure TForm1.N4Click(Sender: TObject);
begin
    Form5.Show //Об авторе
end;

procedure TForm1.N5Click(Sender: TObject);
begin
    if not W3
    then
        ShowMessage('Выполните команду "Запись результатов в файл"')
    else
        Close; //Окончание работы программы
end;

```

Рисунок 11 – Исполняемая часть модуля *Unit1*

Первые четыре процедуры содержат по одному оператору, который вызывает соответствующую форму.

Пятая процедура выполняется при выборе пользователем команды *Выход*. Эта процедура осуществляет окончание работы проекта, только если пользователь ранее (при работе с окном приложения *Form4*) записал в файл результаты расчета.

4.4.2. Модуль *Unit2*

Модуль *Unit2* содержит две процедуры *TForm2.FormActivate* и *TForm2.Button1Click*.

На рисунке 12 приведена схема алгоритма процедуры *TForm2.FormActivate*, вызываемой при активации окна *Form2*.

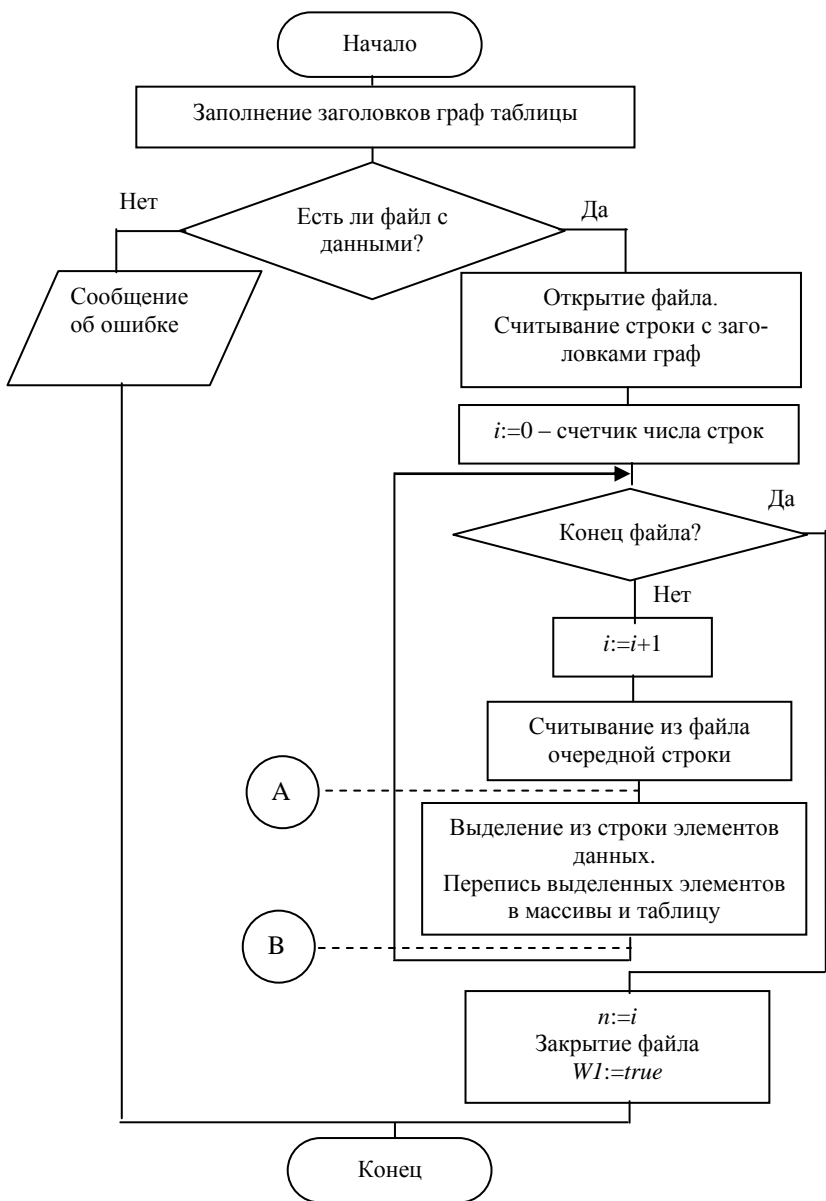


Рисунок 12 – Укрупненная схема алгоритма активации окна *Form2*

На схеме рисунка 12 детализации требует только блок *AB*. Соответствующая схема алгоритма приведена на рисунке 13.

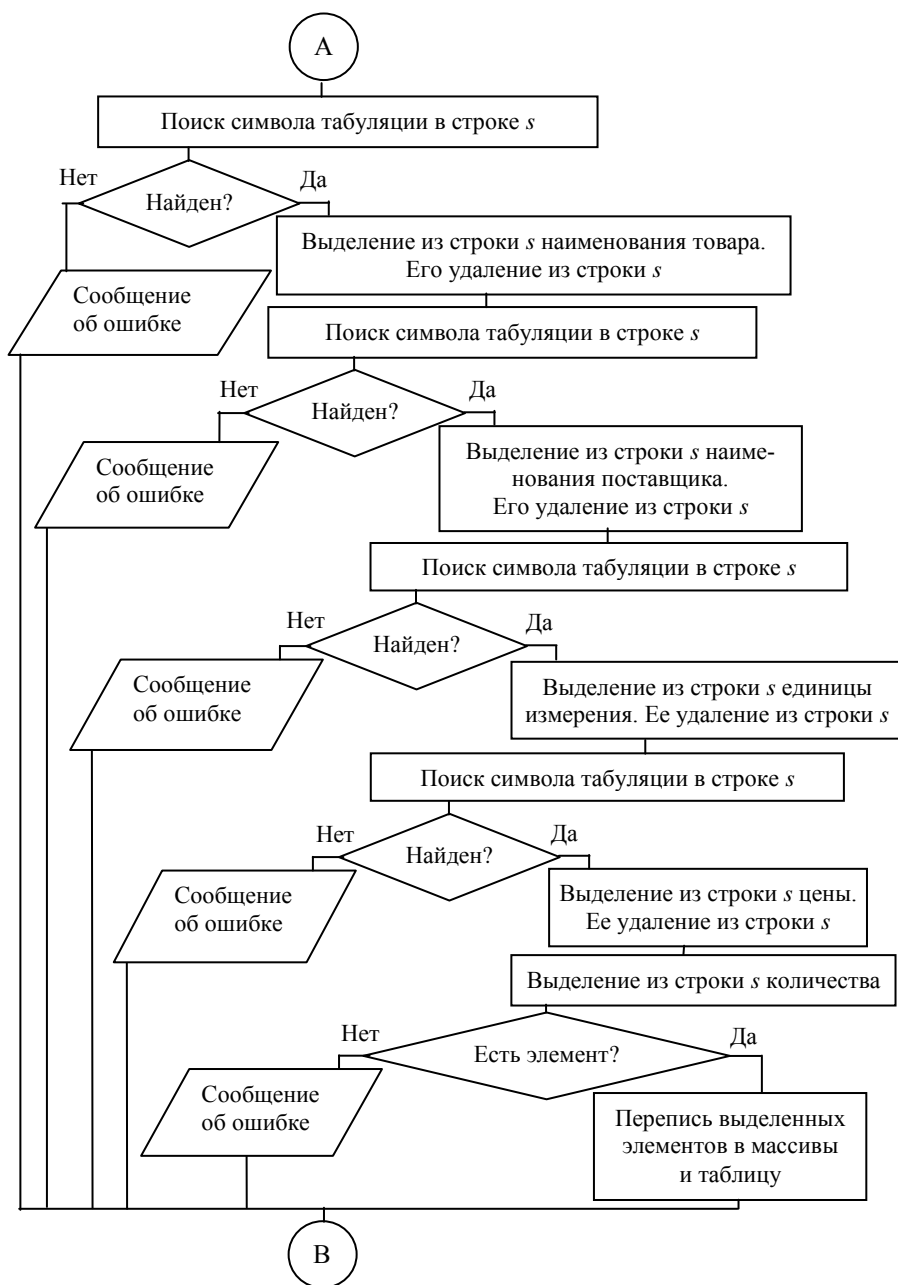


Рисунок 13 – Детализация блока АВ рисунка 12

Текст процедуры *TForm2.FormActivate* приведен на рисунках 14 и 15.

```
procedure TForm2.FormActivate(Sender: TObject);
{Ввод исходных данных из текстового файла,
помещение их в массивы и отображение в таблице строк}
var
  fv1: TextFile;
  s,s1,s2,s3,s4,s5: string[255];
  s6:string[2];
  i,k:integer;
begin
  //Заполнение заголовков граф таблицы
  StringGrid1.Cells[0,0]:='Наименование товара';
  StringGrid1.Cells[1,0]:='Наименование поставщика';
  StringGrid1.Cells[2,0]:='Единица измерения';
  StringGrid1.Cells[3,0]:='Цена, р.';
  StringGrid1.Cells[4,0]:='Количество';
  //Проверка существования файла с исходными данными
  AssignFile(fv1,'Поставки товаров.txt');
  if not FileExists('Поставки товаров.txt')
  then
    ShowMessage('Файла с указанным именем не существует')
  else
    begin
      Reset(fv1); //Открытие файла
      //Считывание первой строки с заголовками граф таблицы
      Readln(fv1,s);
      i:=0; //Счетчик числа строк с данными
      {Цикл считывания строк из файла и перепись информации
      в таблицу StringGrid1 и массивы tov, post, ediz, cena, kol}
      while not (SeekEof(fv1)) do
        begin
          //Считывание очередной строки из файла
          i:=i+1; //Номер считываемой строки с данными
          Str(i:2,s6);
          Readln(fv1,s);
          //Выделение из строки наименования товара
          k:=Pos(Chr($09),s); //Поиск символа табуляции
          if k=0
          then
            ShowMessage('Строка № '+s6+' содержит ошибку')
          else
            begin
              s1:=Copy(s,1,k-1); //Наименование товара
              Delete(s,1,k); //Удаление из s наименования товара
```

Рисунок 14 – Текст процедуры *TForm2.FormActivate* (первая часть)

```

//Выделение из строки наименования поставщика
k:=Pos(Chr($09),s); //Поиск символа табуляции
if (k=0) or (s1='')
then
  ShowMessage('Строка № '+s6+' содержит ошибку')
else
begin
  s2:=Copy(s,1,k-1); //Наименование поставщика
  Delete(s,1,k); //Удаление из s наименования поставщика
  //Выделение из строки единицы измерения
  k:=Pos(Chr($09),s); //Поиск символа табуляции
  if (k=0) or (s2='')
  then
    ShowMessage('Строка № '+ s6+' содержит ошибку')
  else
  begin
    s3:=Copy(s,1,k-1); //Единица измерения
    Delete(s,1,k); //Удаление из s единицы измерения
    //Выделение из строки цены
    k:=Pos(Chr($09),s); //Поиск символа табуляции
    if (k=0) or (s3='')
    then
      ShowMessage('Строка № '+s6+' содержит ошибку')
    else
    begin
      s4:=Copy(s,1,k-1); //Цена
      Delete(s,1,k); //Удаление из s цены
      s5:=s; //Количество
      if (s4='') or (s5='')
      then
        ShowMessage('Строка № '+s6+' содержит ошибку')
      else
      begin
        {Перепись выделенных элементов
         в массивы и таблицу}
        tov[i]:=s1;
        StringGrid1.Cells[0,i]:=s1;
        post[i]:=s2;
        StringGrid1.Cells[1,i]:=s2;
        ediz[i]:=s3;
        StringGrid1.Cells[2,i]:=s3;
        cenafil:=StrToFloat(s4);
        StringGrid1.Cells[3,i]:=s4;
        kol[i]:=StrToFloat(s5);
        StringGrid1.Cells[4,i]:=s5;
      end;
    end;
  end;
end;
end;
n:=i; //Количество записей в файле (без строки с заголовками)
CloseFile(fv1);
W1:=True; //Признак выполнения команды "Ввод исходных данных"
end;
end;

```

Рисунок 15 – Текст процедуры *TForm2.FormActivate*
(заключительная часть)

Отметим, что при выделении элементов данных из строки файла проверяется наличие всех пяти элементов. При отсутствии любого из них выдается сообщение об ошибке с указанием номера строки файла, в котором обнаружена ошибка.

Для поиска в строке позиции табуляции использовалась функция *Chr(\$09)*, которая преобразует шестнадцатеричный код табуляции в литеру.

Вторая процедура модуля *Unit2* обеспечивает закрытие окна формы щелчком по кнопке *В главное меню*, ее текст приведен на рисунке 16.

```
procedure TForm2.Button1Click(Sender: TObject);  
begin  
    Close //В главное меню  
end;
```

Рисунок 16 – Процедура закрытия окна *Form2*

4.4.3. Модуль *Unit3*

Модуль *Unit3* содержит две процедуры *TForm3.FormActivate* и *TForm3.Button1Click*.

Процедура *TForm3.Button1Click* вызывается щелчком по кнопке *В главное меню*. Как и в модуле *Unit2*, эта процедура обеспечивает закрытие окна формы, но вместе с тем она чистит таблицу с результатами сортировки. Текст процедуры приведен на рисунке 17.

```
procedure TForm3.Button1Click(Sender: TObject);  
var  
    i, j: Integer;  
begin  
    for i:=1 to StringGrid1.RowCount-1 do //Чистка таблицы  
        for j:=0 to 4 do  
            StringGrid1.Cells[j, i] := '';  
        Close //В главное меню  
    end;
```

Рисунок 17 – Процедура закрытия окна *Form3*

Процедура *TForm3.FormActivate* использует массивы *tov*, *post*, *ediz*, *cena* и *kol*, а также переменную *n* (количество элементов в каждом массиве), описанные в модуле *Unit1* и определенные в модуле *Unit2*. Схема алгоритма процедуры, вызываемой при активации окна *Form3*, приведена на рисунке 18.



Рисунок 18 – Схема алгоритма процедуры *TForm3.FormActivate*

Сортировка массива поставщиков реализована на основе «метода пузырька», который состоит в следующем. Исходный массив просматривается слева направо с проверкой условия $post[i] > post[i+1]$. Если для какой-то пары элементов это условие выполняется, то данные элементы переставляются и проверка продолжается. После проверки последней пары на последнем месте окажется самый большой элемент (в нашем случае – последний по алфавиту). Затем проверки начинаются опять с первой пары, но последняя пара элементов уже не проверяется. Всего делается $n-1$ просмотров массива.

Отметим, что при перестановке элементов $post[i]$ и $post[i+1]$ переставляются и соответствующие элементы в остальных четырех массивах.

Текст процедуры *TForm3.FormActivate* приведен на рисунке 19.

```
procedure TForm3.FormActivate(Sender: TObject);
{Сортировка массивов исходных данных и
 отображение результата в таблице строк}
var
  s: string[255];
  a: real;
  i, k: integer;
begin
  if not W1
  then
    ShowMessage('Выполните команду "Ввод исходных данных"')
  else
    begin
      //Заполнение заголовков граф таблицы
      StringGrid1.Cells[0,0]:='Наименование товара';
      StringGrid1.Cells[1,0]:='Наименование поставщика';
      StringGrid1.Cells[2,0]:='Единица измерения';
      StringGrid1.Cells[3,0]:='Цена, р.';
      StringGrid1.Cells[4,0]:='Количество';
      {Сортировка массива поставщиков по алфавиту и
      соответствующие перестановки в остальных четырех массивах}
      for k:=n-1 downto 1 do
        for i:=1 to k do
          if post[i]>post[i+1]
          then
            begin
              //Перестановка двух элементов в каждом из 5 массивов
              s:=post[i]; post[i]:=post[i+1]; post[i+1]:=s;
              s:=tov[i]; tov[i]:=tov[i+1]; tov[i+1]:=s;
              s:=ediz[i]; ediz[i]:=ediz[i+1]; ediz[i+1]:=s;
              a:=cena[i]; cena[i]:=cena[i+1]; cena[i+1]:=a;
              a:=kol[i]; kol[i]:=kol[i+1]; kol[i+1]:=a;
            end;
          //Вывод отсортированных массивов в таблицу
        for i:=1 to n do
          begin
            StringGrid1.Cells[0,i]:=tov[i];
            StringGrid1.Cells[1,i]:=post[i];
            StringGrid1.Cells[2,i]:=ediz[i];
            StringGrid1.Cells[3,i]:=FloatToStr(cena[i]);
            StringGrid1.Cells[4,i]:=FloatToStr(kol[i]);
          end;
        W2:=True; //Признак выполнения команды "Сортировка"
      end;
    end;
end;
```

Рисунок 19 – Текст процедуры *TForm3.FormActivate*

4.4.4. Модуль Unit4

Модуль *Unit4* содержит три процедуры *TForm4.FormActivate*, *TForm4.Button1Click* и *TForm4.Button2Click*.

Процедура *TForm4.Button1Click* вызывается щелчком по кнопке *В главное меню*. Как и в модуле *Unit3*, эта процедура обеспечивает закрытие окна формы и чистку таблицы с результатами обработки. Текст процедуры приведен на рисунке 20.

```
procedure TForm4.Button1Click(Sender: TObject);
var
    i: Integer;
begin
    for i:=1 to StringGrid1.RowCount-1 do //Чистка таблицы
    begin
        StringGrid1.Cells[0,i]:='';
        StringGrid1.Cells[1,i]:='';
    end;
    Close //В главное меню
end;
```

Рисунок 20 – Процедура закрытия окна *Form4*

В интерфейсной части модуля *Unit4* описаны два массива: массив строк *postraz*, предназначенный для хранения различных имен поставщиков, и вещественный массив *stoim*, в который будут занесены вычисленные стоимости поставок каждого поставщика. Кроме того, описана целочисленная переменная *kp*, значение которой – количество различных поставщиков. Все эти объекты получают значения в процедуре *TForm4.FormActivate* и используются в процедуре *TForm4.Button2Click* (команда *Запись результатов в файл*).

Активация окна приложения *Form4*, а следовательно, и выполнение процедуры *TForm4.FormActivate* происходят после выбора пользователем в окне *Form1* команды *Обработка*. Процедура анализирует, выполнены ли ранее команды *Ввод исходных данных* и *Сортировка*. Если какая-то из данных команд не выполнена, то выдается сообщение пользователю и выполнение процедуры на этом завершается. Если обе команды были ранее выполнены, то организуется цикл просмотра отсортированного массива *post* с вычислением для каждого элемента стоимости поставки (произведение цены на количество) и добавление полученного значения к накапливаемой сумме. Имя поставщика и накопленная сумма переписываются в массивы *postraz* и *stoim*. При этом значение переменной *kp* увеличивается на 1.

Схема алгоритма процедуры *TForm4.FormActivate* приведена на рисунке 21.

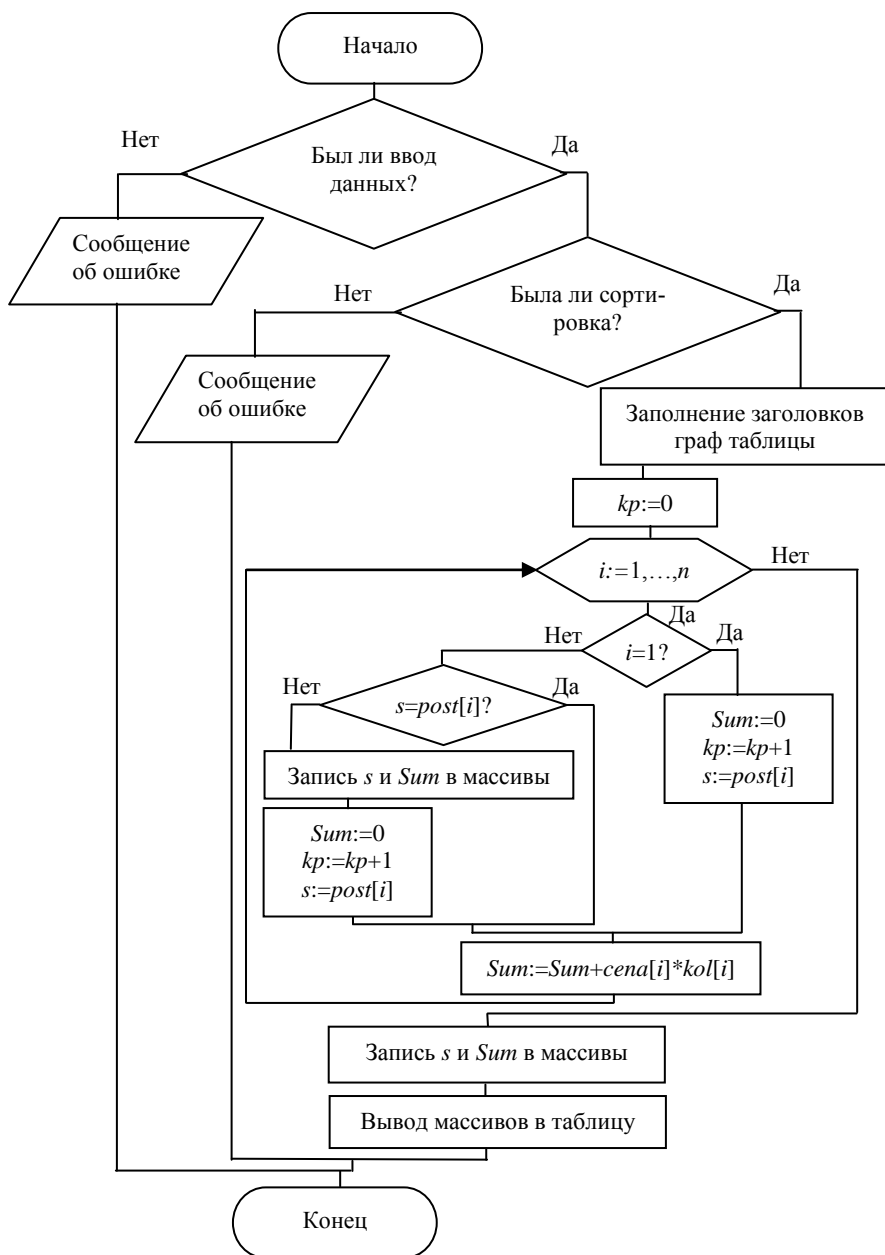


Рисунок 21 – Схема алгоритма процедуры *TForm4.FormActivate*

Текст процедуры *TForm4.FormActivate* приведен на рисунках 22 и 23.

```
procedure TForm4.FormActivate(Sender: TObject);
//Обработка отсортированной информации
var
  s:string[255];
  Sum:real;
  i:integer;
begin
  if not W1
  then
    ShowMessage('Выполните команду "Ввод исходных данных"')
  else
    if not W2
    then
      ShowMessage('Выполните команду "Сортировка"')
    else
      {Ранее были выполнены команды
      "Ввод исходных данных" и "Сортировка"}
      begin
        //Заполнение заголовков граф таблицы
        StringGrid1.Cells[0,0]:='Наименование поставщика';
        StringGrid1.Cells[1,0]:='Стоимость, р.';
        kp:=0; //Количество различных поставщиков
        for i:=1 to n do
          begin
            if i=1
            then
              begin
                Sum:=0;
                kp:=kp+1;
                s:=post[i]; //Запомнили имя первого поставщика
              end
            else
              if s<>post[i]
              then //Встретился новый поставщик
                begin
                  //Занесение в массив наименование поставщика
                  postraz[kp]:=s;
                  //Занесение в массив общей стоимости
                  stoim[kp]:=Sum;
                  Sum:=0;
                  kp:=kp+1;
                  s:=post[i]; //Запомнили имя нового поставщика
                end;
              //Накопление стоимости для одного поставщика
              Sum:=Sum+cena[i]*kol[i]
            end;
          end;
```

Рисунок 22 – Текст процедуры *TForm4.FormActivate* (первая часть)

```

// Записывание в массивы данных о последнем поставщике
postraz[kp] := s;
stoim[kp] := Sum;
// Вывод результатов в таблицу
for i:=1 to kp do
begin
StringGrid1.Cells[0,i] := postraz[i];
StringGrid1.Cells[1,i] := FloatToStr(stoim[i]);
end;
end;
end;

```

Рисунок 23 – Текст процедуры *TForm4.FormActivate*
(заключительная часть)

Процедура *TForm4.Button2Click* выполняется щелчком по кнопке *Запись результатов в файл* в окне приложения *Form4*. Назначение процедуры – запись в файл *Общая стоимость поставок.txt* содержимого массивов *postraz* и *stoim*.

Схема алгоритма процедуры *TForm4.Button2Click* приведена на рисунке 24, а текст процедуры – на рисунке 25.

4.5. Результат выполнения проекта

Для исходных данных, представленных на рисунке 2, получен файл *Общая стоимость поставок.txt* (рисунок 26). Содержимое полученного файла полностью согласуется с тестом, приведенным на рисунке 3.

4.6. Направления развития проекта

Сервис проекта существенно улучшится, если в него добавить по крайней мере две следующие функции:

- обработка исключительных ситуаций, т. е. перехват программой прерываний среды Delphi при обнаружении ошибки, информирование об ошибке пользователя и предоставление ему возможности внести исправление;
- использование визуальных компонентов *OpenDialog* и *SaveDialog*, реализующих стандартные диалоговые окна открытия и сохранения файла, с помощью которых можно будет указывать пути к исходному и результирующему файлам.

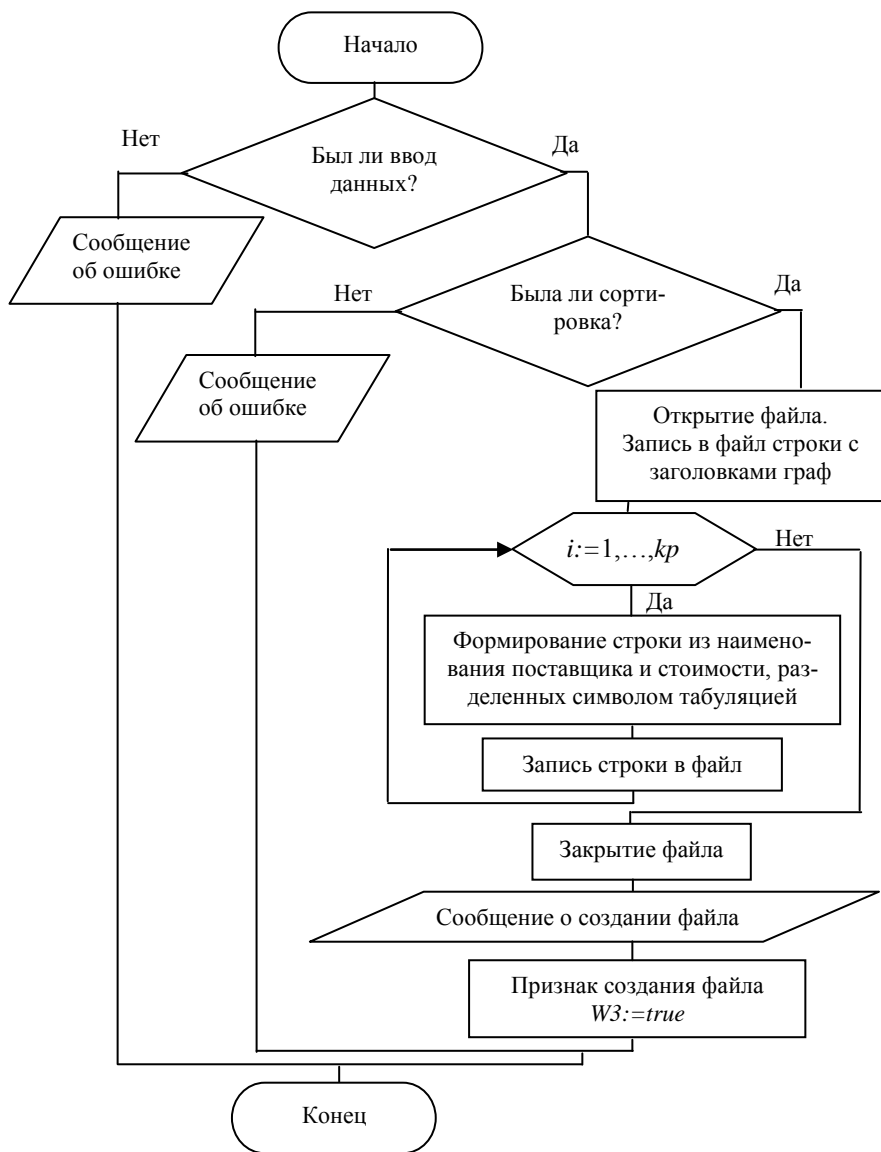


Рисунок 24 – Схема алгоритма процедуры *TForm4.Button2Click*

```

procedure TForm4.Button2Click(Sender: TObject);
//Запись результатов в новый текстовый файл
var
    i:Integer;
    s:string[255];
begin
    if not W1
    then
        ShowMessage('Выполните команду "Ввод исходных данных"')
    else
        if not W2
        then
            ShowMessage('Выполните команду "Сортировка"')
        else
            {Ранее были выполнены команды
             "Ввод исходных данных" и "Сортировка"}
            begin
                Rewrite(fv2); //Открытие нового файла
                //Запись в файл строки с заголовками граф
                Writeln(fv2,'Наименование поставщика  Стоимость, р. ');
                for i:=1 to kp do // Цикл записи в файл строк таблицы
                begin
                    //Преобразование значения стоимости в текстовый формат
                    Str(stoim[i]:12:0,s);
                    {Формирование строки из наименования поставщика и
                     стоимости, разделенных символом табуляции}
                    s:=postraz[i]+' '+ s; //
                    Writeln(fv2,s); //Запись в файл очередной строки
                end;
                CloseFile(fv2); //Закрытие файла
                ShowMessage('Информация из таблицы записана'+
                    ' в файл "Общая стоимость поставок.txt"');
                W3:=True; //Признак создания нового файла
            end;
    end;

```

Рисунок 25 – Текст процедуры *TForm4.Button2Click*

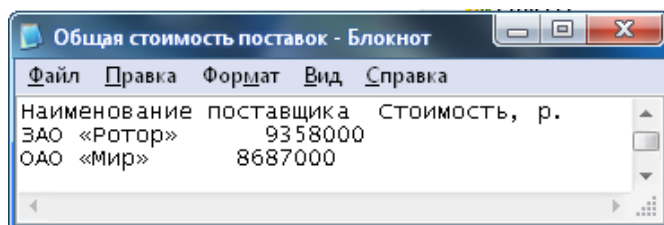


Рисунок 26 – Результат тестирования проекта

4.7. Инструкция пользователя

4.7.1. Назначение программы

Данная программа считывает с диска текстовый файл *Поставки товаров.txt*, в котором находится ведомость, содержащая данные о поставках товаров: *Наименование товара*; *Наименование поставщика*; *Единица измерения*; *Цена, р.*; *Количество*. Известно, что один поставщик может поставлять несколько разных товаров, и каждый товар может быть неоднократно поставлен разными фирмами. После считывания производится сортировка файла по полю *Наименование поставщика* (по алфавиту), а затем вычисление общей стоимости поставок каждого поставщика. Результаты расчета записываются в новый текстовый файл *Общая стоимость поставок.txt*, содержащий таблицу из двух граф: *Наименование поставщика*; *Стоимость, р.*

4.7.2. Условия выполнения программы

Исполняемый файл *Project1.exe* занимает 747 Кб памяти и может выполняться на любом персональном компьютере под управлением операционной системы Windows версии не ниже Windows 4.00.950 (Windows 95).

Результат выполнения записывается на жесткий магнитный диск и отображается на экране монитора (принтер не используется).

Текстовый файл с исходными данными должен находиться в той же папке, что и исполняемый файл. Результирующий файл помещается в ту же папку.

Исходные файлы проекта (24 файла) занимают на диске 1,15 Мб. Для выполнения программы эти файлы не нужны. Они необходимы программисту для внесения изменений в проект.

4.7.3. Выполнение программы

Перед выполнением программы надо создать текстовый файл *Поставки товаров.txt*, пример которого представлен на рисунке 2. Элементы данных в каждой строке файла должны разделяться символом табуляции. Первой строкой файла должна быть строка с заголовками граф.

После запуска программы на экране появится главное окно приложения, представленное на рисунке 5. Первой должна быть выполнена команда *Ввод исходных данных*. В результате появится окно приложения, пример которого представлен на рисунке 27.

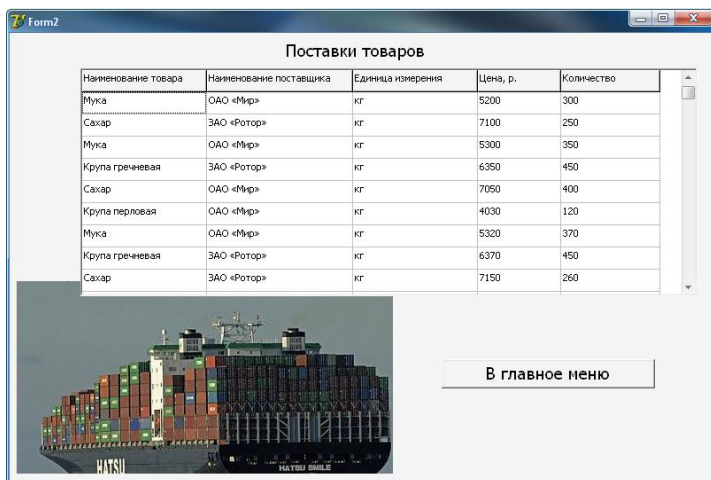


Рисунок 27 – Пример окна приложения после выполнения команды *Ввод исходных данных*

Для продолжения работы необходимо выполнить команду *В главное меню*, после которой на экране появится главное окно приложения.

Далее следует выполнить команду *Сортировка*. В результате появится окно приложения, пример которого представлен на рисунке 28.

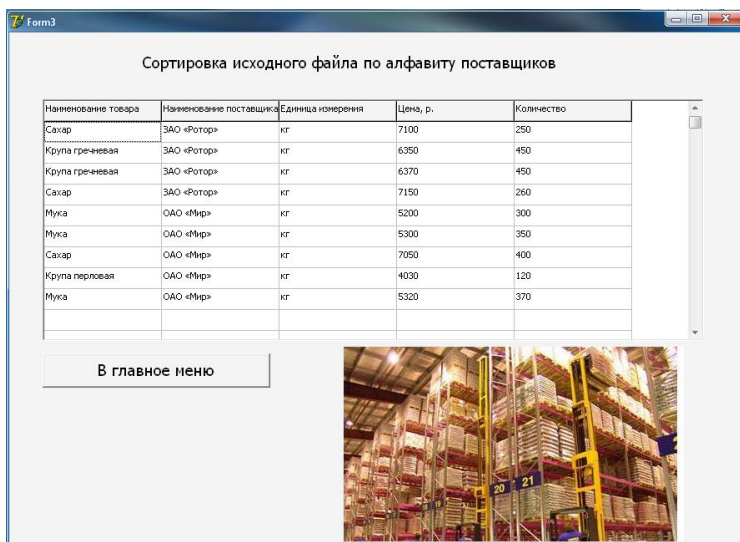


Рисунок 28 – Пример окна приложения после выполнения команды *Сортировка*
 Для продолжения работы необходимо выполнить команду *В главное меню*, после которой на экране появится главное окно приложения.

Следующей выполняется команда *Обработка*. Результат выполнения этой команды отображается в окне приложения, пример которого представлен на рисунке 29.

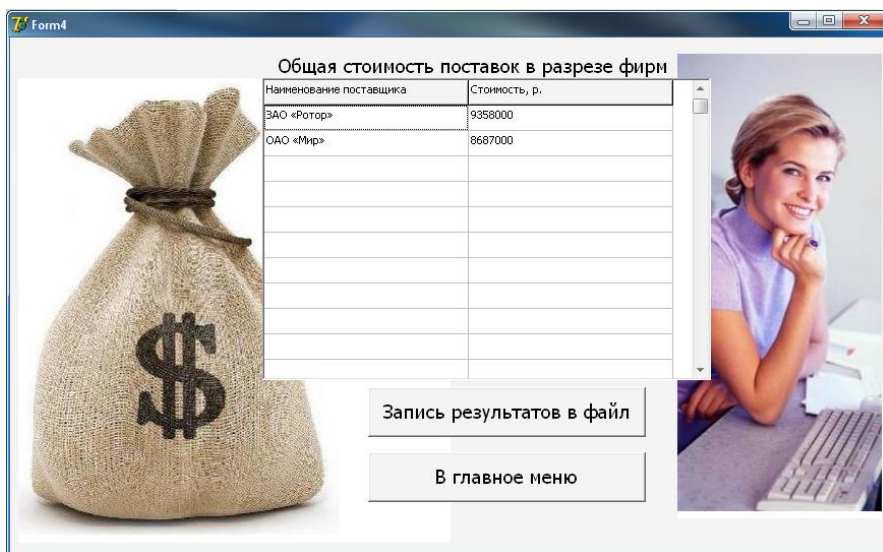


Рисунок 29 – Пример окна приложения после выполнения команды *Обработка*

Естественным продолжением работы является выполнение команды *Запись результатов в файл*. В результате будет создан искомый текстовый файл *Общая стоимость поставок.txt*. Элементы данных в каждой строке файла разделяются символом табуляции. Первой строкой файла будет строка с заголовками граф. Пример выходного файла приведен на рисунке 26.

Выполнение команды *В главное меню* приводит к появлению на экране главного окна приложения.

Команда *Об авторе* инициирует появление окна, содержащего сведения о разработчике проекта. Данная команда может выполняться в любой момент.

Команда *Выход* завершает выполнение программы.

4.7.4. Сообщения пользователю

Ниже перечислены все сообщения, выдаваемые во время выполнения программы.

При выполнении команды *Ввод исходных данных* может быть выдано сообщение, приведенное на рисунке 30, если файл с исходными данными не существует или находится в другой папке.

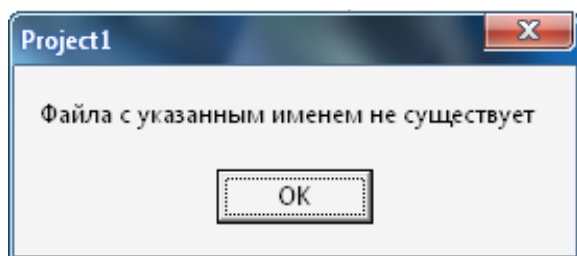


Рисунок 30 – Сообщение об отсутствии файла исходных данных

В процессе считывания записей из исходного файла может быть выдано сообщение, пример которого приведен на рисунке 31. Данное сообщение выдается в том случае, когда очередная строка исходного файла содержит менее пяти элементов данных.

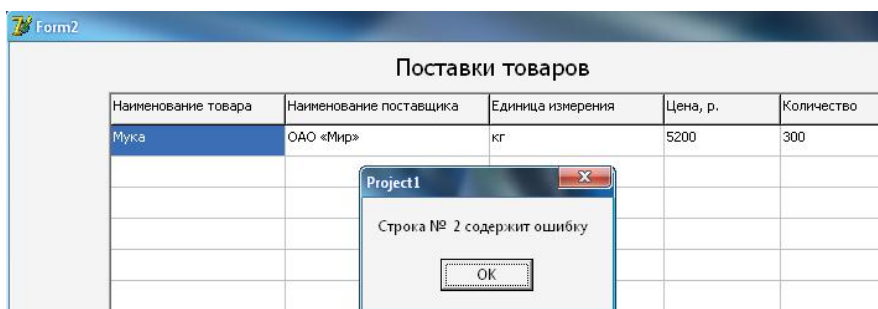


Рисунок 31 – Сообщение о присутствии ошибки в строке файла исходных данных

При попытке выполнения одной из команд *Сортировка*, *Обработка* или *Запись результатов в файл* перед командой *Ввод исходных данных* программа выводит сообщение, представленное на рисунке 32.

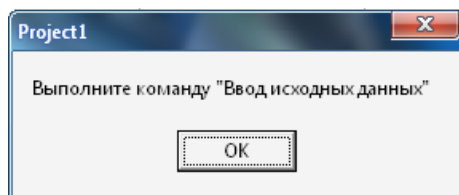


Рисунок 32 – Сообщение о необходимости выполнения команды *Ввод исходных данных*

При попытке выполнения одной из команд *Обработка* или *Запись результатов в файл* перед командой *Сортировка* программа выводит сообщение, представленное на рисунке 33.

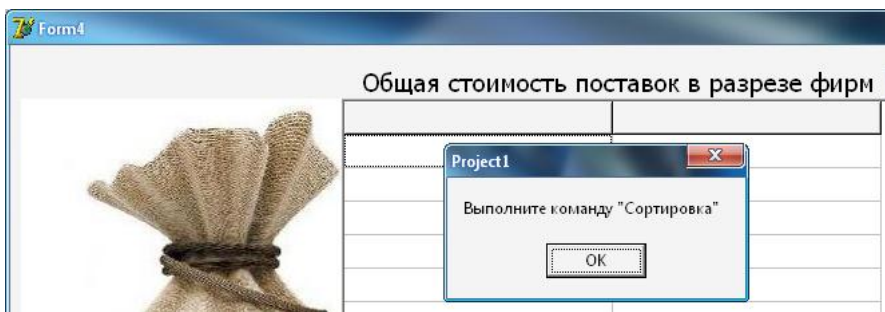


Рисунок 33 – Сообщение о необходимости выполнения команды *Сортировка*

После успешного выполнения команды *Запись результатов в файл* выводится сообщение, представленное на рисунке 34.

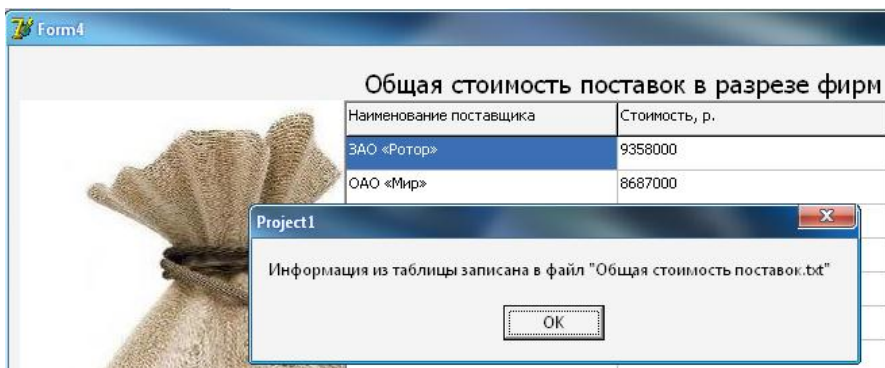


Рисунок 34 – Сообщение об успешном создании выходного файла

При попытке завершить работу программы без записи результатов в файл выводится сообщение, приведенное на рисунке 35.

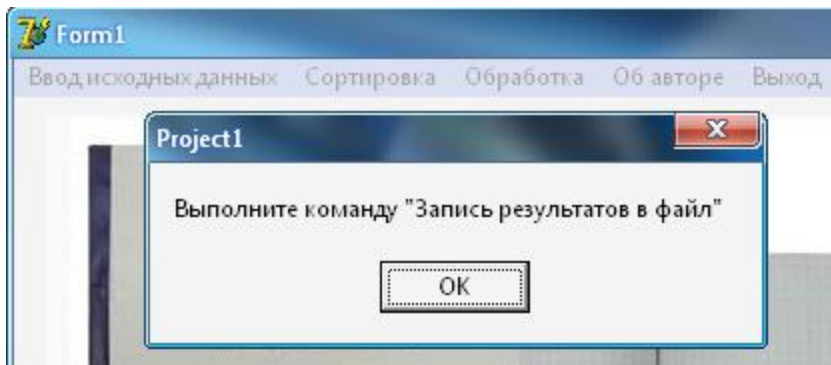


Рисунок 35 – Сообщение о необходимости выполнения команды
Запись результатов в файл

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

Попов, В. Б. Паскаль и Дельфи. Самоучитель / В. Б. Попов. – СПб. : Питер, 2005. – 576 с.

Алексеев, В. Е. Вычислительная техника и программирование. Практикум по программированию : практ. пособие / В. Е. Алексеев, А. С. Ваулин, Г. Б. Петрова ; под ред. А. В. Петрова. – М. : Высш. шк., 1991. – 399 с.

Алгоритмизация и программирование : пособие / авт.-сост. : С. М. Мовшович, О. А. Кравченко. – Гомель : Бел. торгово-экон. ун-т потребит. кооп., 2008. – 104 с.

Обработка экономической информации в системе программирования Delphi : пособие / авт.-сост. : С. М. Мовшович, О. А. Кравченко. – Гомель : Бел. торгово-экон. ун-т потребит. кооп., 2011. – 128 с.

Технологии программирования : практикум / авт.-сост. : Е. В. Коробейникова, С. М. Мовшович, О. А. Кравченко. – Гомель : Бел. торгово-экон. ун-т потребит. кооп., 2012. – 100 с.

Фаронов, В. В. Turbo Паскаль 7.0. Начальный курс : учеб. пособие / В. В. Фаронов. – М. : КноРус, 2006. – 576 с.

Фаронов, В. В. Delphi. Программирование на языке высокого уровня : учеб. / В. В. Фаронов. – СПб. : Питер, 2004. – 640 с.

Глинский, Я. В. Turbo Pascal 7.0. Delphi : учеб. пособие / Я. В. Глинский, В. Е. Анохин, В. А. Ряжская. – СПб. : ДиаСофтЮП, 2001. – 208 с.

Дарахвелидзе, П. Программирование в Delphi 5 : учеб. пособие / П. Дарахвелидзе. – СПб. : BHV, 2000. – 767 с.

Культин, Н. Основы программирования в Delphi : учеб. пособие / Н. Культин. – СПб. : BHV, 2003. – 566 с.

Гофман, В. Delphi 6. Наиболее полное руководство / В. Гофман. – СПб. : BHV, 2001. – 1135 с.

Корняков, В. Программирование документов MS Office в Delphi / В. Корняков. – СПб. : BHV, 2005. – 489 с.

Могилев, А. В. Информатика : учеб. пособие / А. В. Могилев, Н. И. Пак, Е. К. Хеннер ; под ред. Е. К. Хеннера. – М. : Академия, 1999. – 816 с.

Меняев, М. Ф. Информатика и основы программирования : учеб. пособие / М. Ф. Меняев. – М. : Омега-Л, 2005. – 432 с.

Иллюстрированный самоучитель по Delphi 7 для начинающих [Электронный ресурс]. – Электрон. текстовые и граф. данные (5,2 Мб). – Режим доступа : http://www.inet-knigi.org/index.php?option=com_content&task=view&id=25&Itemid=36. – Дата доступа : 27.12.2012.

32 урока по Delphi [Электронный ресурс]. – Электрон. текстовые данные (1,5 Мб). – Режим доступа : <http://www.mexxs.net/books.php?file=19>. – Дата доступа : 27.12.2012.

Иллюстрированный самоучитель по Delphi 7 для начинающих // Programmers Klondike [Электронный ресурс]. – Режим доступа : http://www.proklondike.com/books/delphi/delphi_illustr_samouchitel_delphi7.html. – Дата доступа : 27.12.2012.

Фленов, М. Библия Delphi / М. Фленов [Электронный ресурс]. – Электрон. текстовые и граф. данные (55,79 Мб). – СПб. : БХВ-Петербург, 2004. – 880 с. – Режим доступа : http://soft-plus.ucoz.ru/load/kniga_biblija_delphi_mikhail_flenov/131-1-0-1090. – Дата доступа : 27.12.2012.

ПРИЛОЖЕНИЕ

Образец оформления титульного листа курсовой работы

БЕЛОРУССКИЙ РЕСПУБЛИКАНСКИЙ СОЮЗ ПОТРЕБИТЕЛЬСКИХ ОБЩЕСТВ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ «БЕЛОРУССКИЙ ТОРГОВО-ЭКОНОМИЧЕСКИЙ УНИВЕРСИТЕТ ПОТРЕБИТЕЛЬСКОЙ КООПЕРАЦИИ»

Кафедра информационно-вычислительных систем

КУРСОВАЯ РАБОТА по дисциплине «АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ»

на тему
«Автоматизация учета стоимости поставок товаров
по поставщикам»
(вариант _____)
(номер)

Студент

_____	_____	_____
(группа)	(подпись)	(фамилия, имя, отчество)

Научный руководитель

_____	_____	_____
(ученая степень, ученое звание, должность)	(подпись)	(фамилия, имя, отчество)

Гомель _____
(год)

СОДЕРЖАНИЕ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА	3
1. ЦЕЛЬ, СОДЕРЖАНИЕ И ЗАЩИТА КУРСОВОЙ РАБОТЫ.....	4
2. СОДЕРЖАНИЕ И ОФОРМЛЕНИЕ КУРСОВОЙ РАБОТЫ	4
2.1. Структурные части курсовой работы	4
2.2. Общие правила оформления	5
2.3. Титульный лист.....	8
2.4. Составление оглавления курсовой работы.....	8
2.5. Постановка задачи. Тест	8
2.6. Функционально-модульная структура проекта	8
2.7. Интерфейс проекта	9
2.8. Описание алгоритмов и тексты процедур	9
2.9. Результат выполнения проекта.....	10
2.10. Инструкция пользователя	10
2.11. Направления развития проекта.....	11
2.12. Список использованных источников.....	11
2.13. Приложения.....	11
3. ВАРИАНТЫ ЗАДАНИЙ	12
4. ПРИМЕР ВЫПОЛНЕНИЯ ПРОЕКТА.....	22
4.1. Постановка задачи. Тест	22
4.2. Функционально-модульная структура проекта	24
4.3. Интерфейс проекта	27
4.4. Описания алгоритмов и тексты процедур	30
4.4.1. Модуль Unit1	30
4.4.2. Модуль Unit2.....	32
4.4.3. Модуль Unit3.....	37
4.4.4. Модуль Unit4.....	40
4.5. Результат выполнения проекта.....	43
4.6. Направления развития проекта.....	43
4.7. Инструкция пользователя	46
4.7.1. Назначение программы	46
4.7.2. Условия выполнения программы.....	46
4.7.3. Выполнение программы.....	46
4.7.4. Сообщения пользователю	49
СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ	52
ПРИЛОЖЕНИЕ.....	54

Учебное издание

АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ

**Пособие
по выполнению курсовых работ
для реализации содержания образовательных программ
высшего образования I ступени**

Автор-составитель
Мовшович Семен Михайлович

Редактор М. П. Любошенко
Технический редактор И. А. Козлова
Компьютерная верстка Л. Г. Макарова

Подписано в печать 11.09.14. Формат 60 × 84 ¹/₁₆.
Бумага типографская № 1. Гарнитура Таймс. Ризография.
Усл. печ. л. 3,25. Уч.-изд. л. 3,30. Тираж 75 экз.
Заказ № 73-09-14.

Издатель и полиграфическое исполнение:
учреждение образования «Белорусский торгово-экономический
университет потребительской кооперации».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/138 от 08.01.2014.
Просп. Октября, 50, 246029, Гомель.
<http://www.i-bteu.by>

**БЕЛКООПСОЮЗ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БЕЛОРУССКИЙ ТОРГОВО-ЭКОНОМИЧЕСКИЙ
УНИВЕРСИТЕТ ПОТРЕБИТЕЛЬСКОЙ КООПЕРАЦИИ»**

Кафедра информационно-вычислительных систем

АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ

**Пособие
по выполнению курсовых работ
для реализации содержания образовательных программ
высшего образования I степени**

Гомель 2014